



# Mobilitás és MANET

## Intelligens közlekedési rendszerek

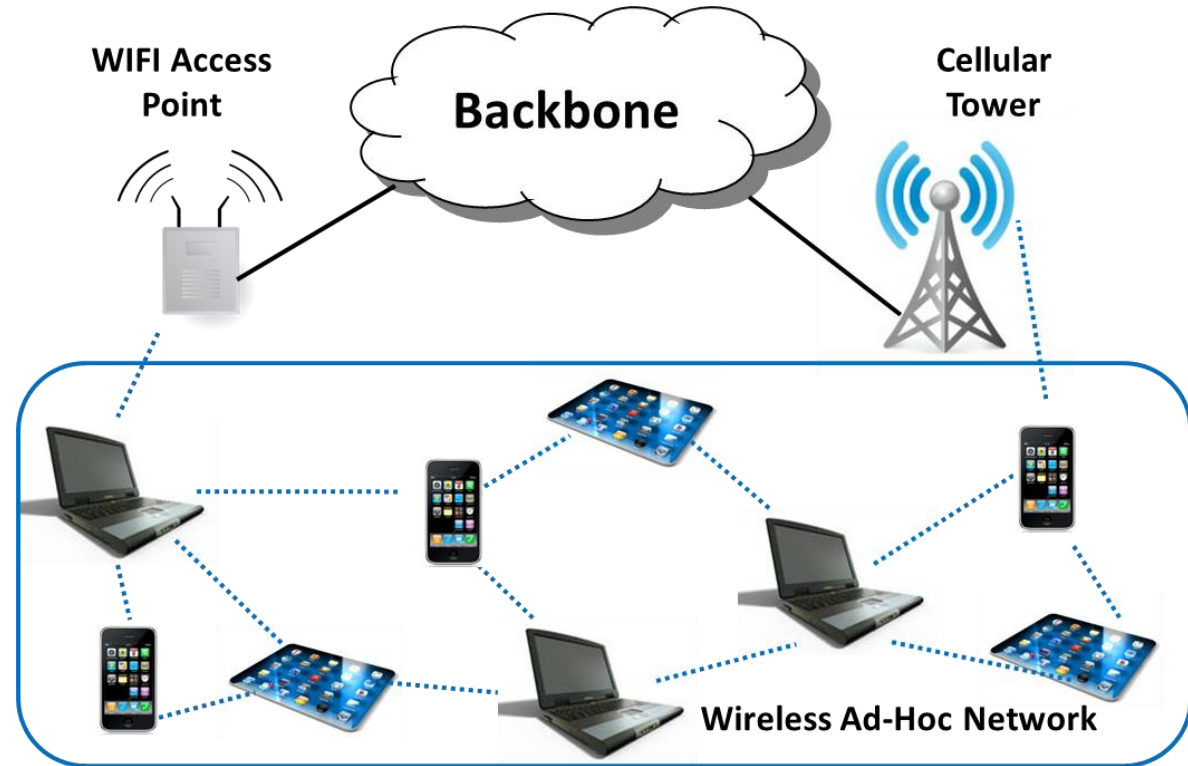
---

VITMMA10 – Okos város MSc mellékspecializáció

Vida Rolland

# Áttekintés

- MANET – Mobile Ad Hoc Networks
- **„Ad Hoc”** jelentése
  - Azonnal, ideiglenesen, előkészület nélkül
  - Ad hoc bizottság = ideiglenes testület, rendkívüli, a szervezet folyamatos működésétől eltérő feladat megoldására



# Ad hoc hálózatok

- **Infrastruktúra-mentes hálózat**

- Internet kapcsolat nélkül
- Előre telepített szerverek (AAA, DHCP, stb.), szolgáltatások nélkül

- **Nincs IP alhálózat alapú címzés**

- Problémát okoz a “klasszikus” routing protokollok számára

- **Nincsenek megbízható (stabil) hálózati eszközök**

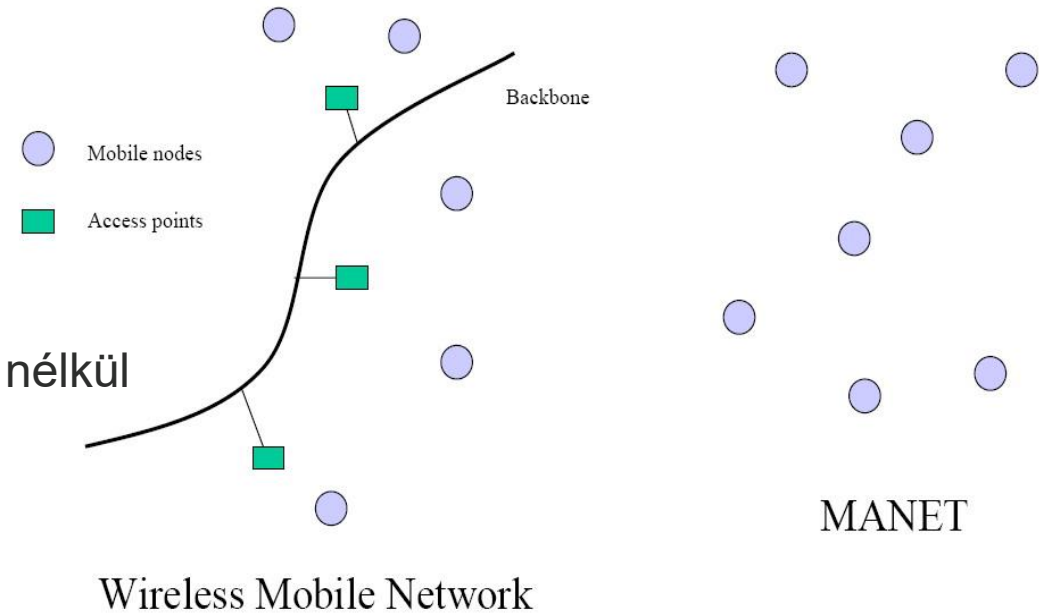
- Bármikor változhat a „szomszédom” állapota – lemerül, eltávolodik, stb.
- Nem ismerem a szomszédomat, nem tudom hogy megbízhatók-e benne

- **Önszerveződés**

- Peer-to-peer paradigma (layer 3, hálózati szinten)

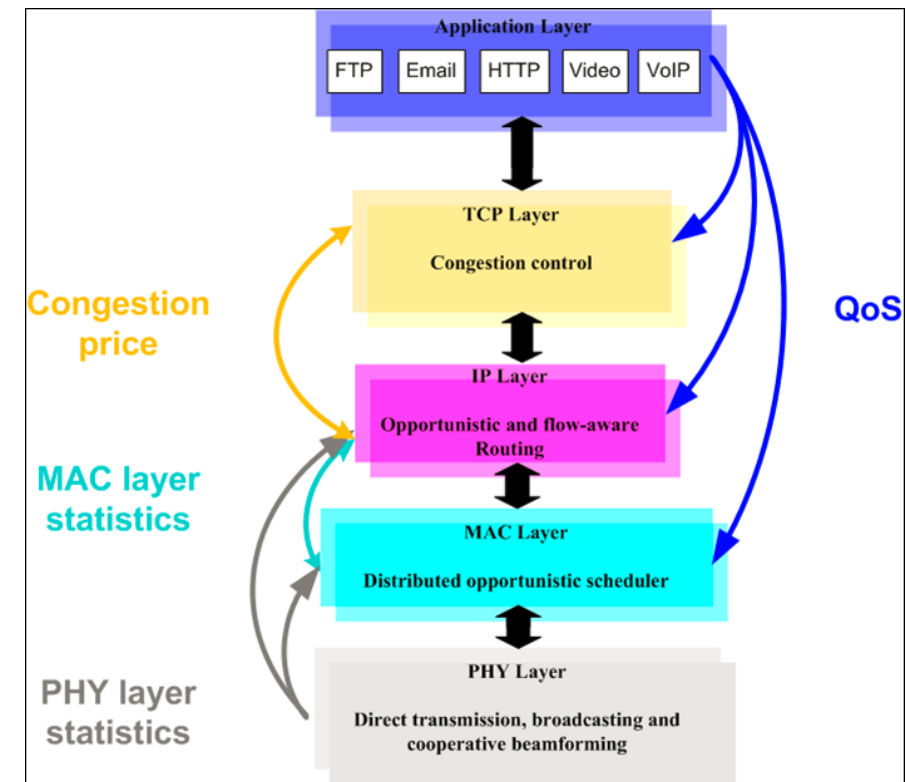
- **Multi-hop**

- kommunikáció (routing) több ugráson (eszközön) keresztül



# MANET kutatási témák

- Fizikai réteg -> „*mobilitás modellek*”
  - Energiatakarékos működés, adaptív rádiós teljesítmény
  - Mobilitáshoz igazított rádiós technológiák
- **Adatkapcsolati réteg**
  - MAC (osztott közeghozzáférés, hatékonyság, ütközések csökkentése)
- **Hálózati réteg**
  - Útválasztás (dinamikus topológia, prefix-alapú routing nem működik)
- Felsőbb rétegek
  - Csomag újraküldés, TCP (csomagvesztés, „bizonytalan” közeg)
  - Biztonság (kiterjeszhető mindegyik rétegre)
- Cross-layer optimization
  - Az ISO/OSI model több rétegének együttes optimalizálása
  - Minden egyes rétegnek van hatása a mobilitásra

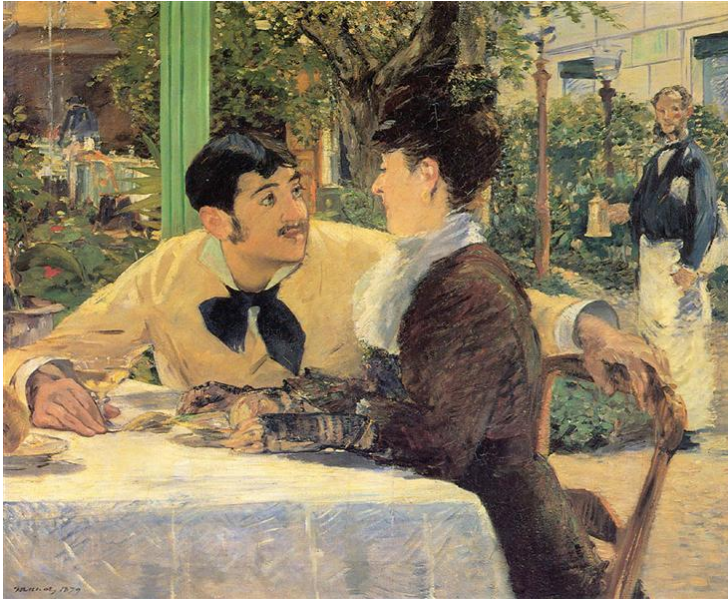
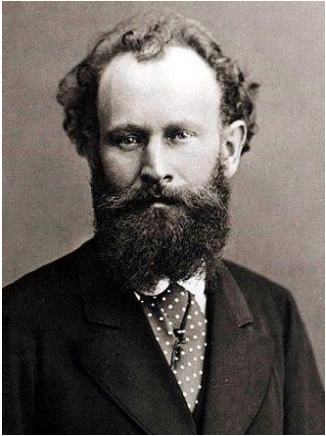


# Mobilitás típusok

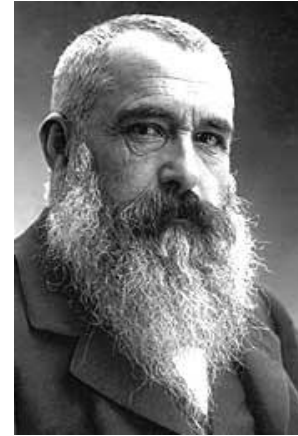
- Nomád (**nomadic**) mobilitás
  - Mozgás során *nincs* aktivitás – kikapcsolt állapot
  - Új cím kiosztása, a régi kapcsolatok felvétele a helyváltoztatás után.
- Lassú (**slow**) mobilitás
  - Pl. konferencia halljában sétáló emberek
  - egyetemi campus – sétáló, bicikliző diákok
  - várótermek
- Gyors (**fast**) mozgás
  - Autók, motorkerékpárok, ...
- Együtt mozgó hálózatok...

# MANET vs. MONET

Edouard MANET



Claude MONET



## Mobile Ad Hoc Network

## Moving Networks

- Együttmozgó hálózatok
  - Pl.: vonat, metró, busz, repülőgép utasai
- Alternatív elnevezés
  - **Networks in Motion – NEMO**

# NEMO – mozgó hálózatok

- Sok MN közös mozgása
  - Ha amúgy is együtt mozognak, kezeljük együtt a mobilitásukat
- MR (mobil router) – default gateway
  - A NEMO-tagok és a külvilág közti kapcsolat biztosítása
  - Dedikált eszköz, vagy egy a sok közül (periodikusan szerepet cserélve)
    - Jellemzően a legtöbb energiával rendelkező akkumulátor, legnagyobb sávszélesség, stb.
- Az MN-eknek regisztrálniuk kell az MR-en
  - Az MR alhálózatához tartoznak
  - “Fix” csomópontok a hálózatban, (MR-hez képest) a relatív helyzetük változatlan
    - Emiatt Fixed Local Node-ként (FLN) is hivatkozzák őket



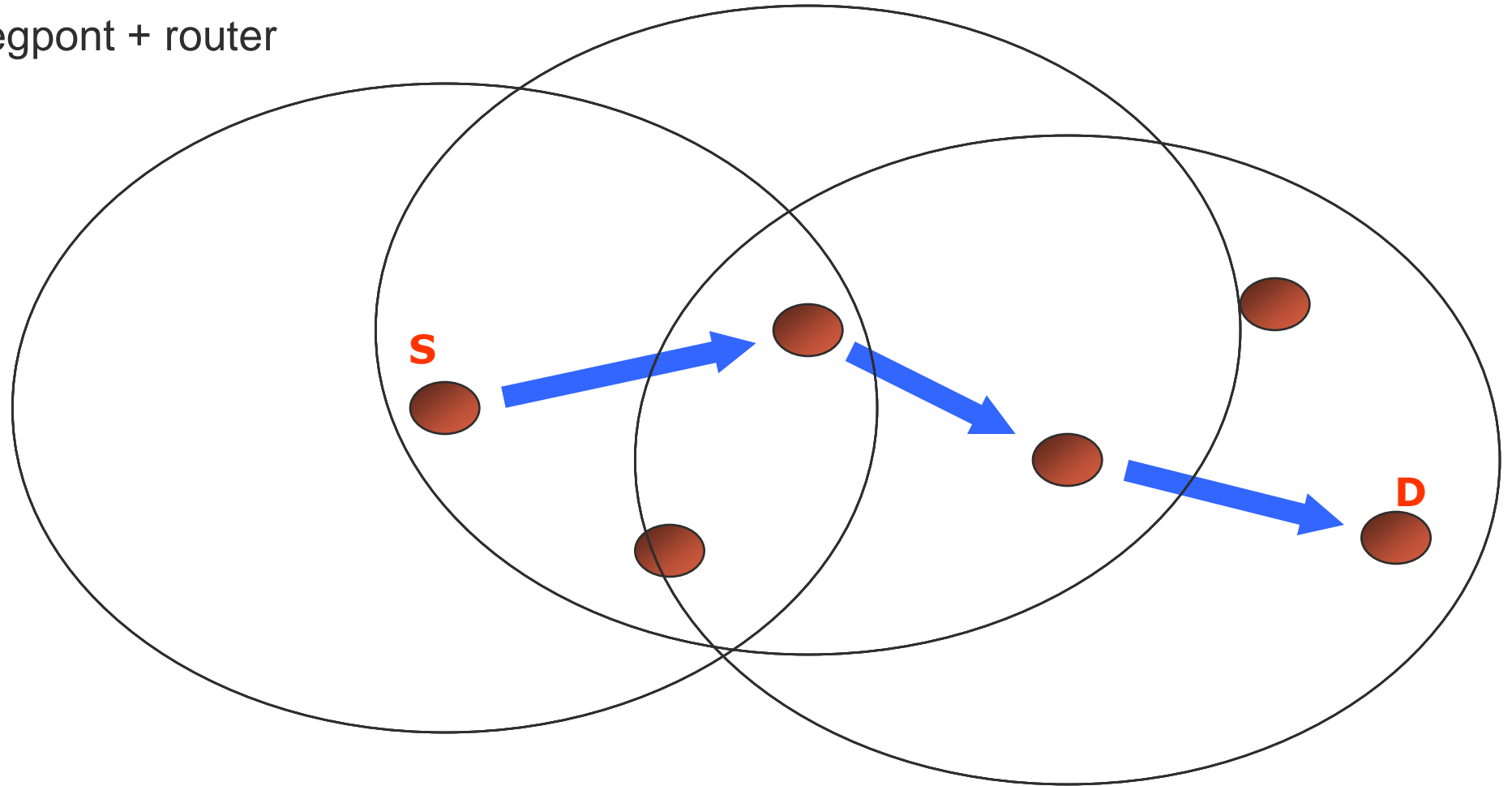
# NEMO hatékonysága függ a környezetétől

- (Lehetséges) hátrány:
  - Pl. 100 MN 3G/4G mobil internet hozzáféréssel egy városi környezetben
  - Ha az MN-ek nem lépnek be a NEMO-ba
    - - egyedi mobilitás menedzsment szükséges mind a 100 MN részére
    - + Egyenként megkapják a technológia által biztosított sávszélességet
  - Ha az MN-ek mind ugyanabba a NEMO-ba lépnek be
    - A szűk keresztmetszet az MR link kapacitása
    - Legrosszabb esetben az MN-ek csak 1/100-ad részét kapják az előző esetben számolt sávszélességnek
- (Lehetséges) előny:
  - Pl. 100 MN egy repülőgépen akar hálózatra csatlakozni
    - Gyakorlatilag az MR az egyetlen lehetőség a kapcsolatra
  - A mobilitás menedzsment optimális
    - Csak az MR mobilitását kell menedzselni



# MANET routing

- Pontról pontra
- Mobil állomás = végpont + router



# Általában az útválasztásról

- Nagyon sok útválasztó protokollt kidolgoztak
  - Egyesek speciálisan a MANET-re
  - Másokat adaptálták a vezetékes hálózatok világából
- Nincs egyetlen protokoll, amely minden helyzetben jól működik
  - Egyes megközelítések ún. **adaptív protokollokat** definiáltak
- Kívánatos tulajdonságok vezeték nélküli ad-hoc routingnál
  - Elosztott működés
  - Hurokmentesség
  - Igény szerinti működés
  - Biztonság
  - „Alvás” periódusok támogatása
  - Egyirányú linkek támogatása

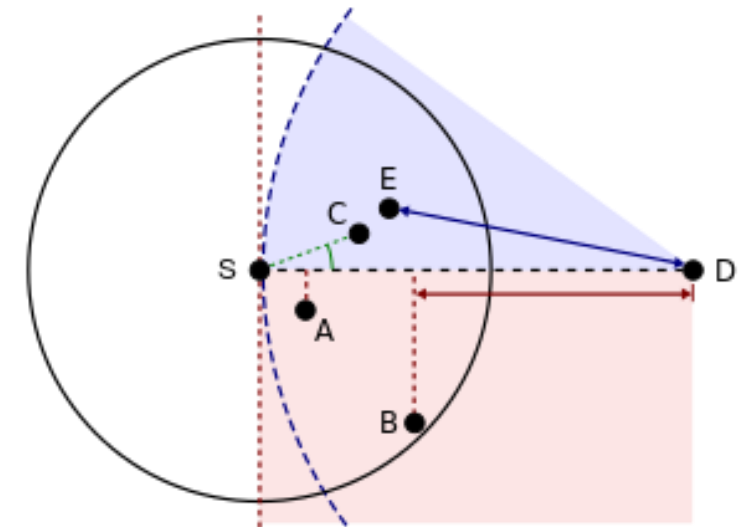
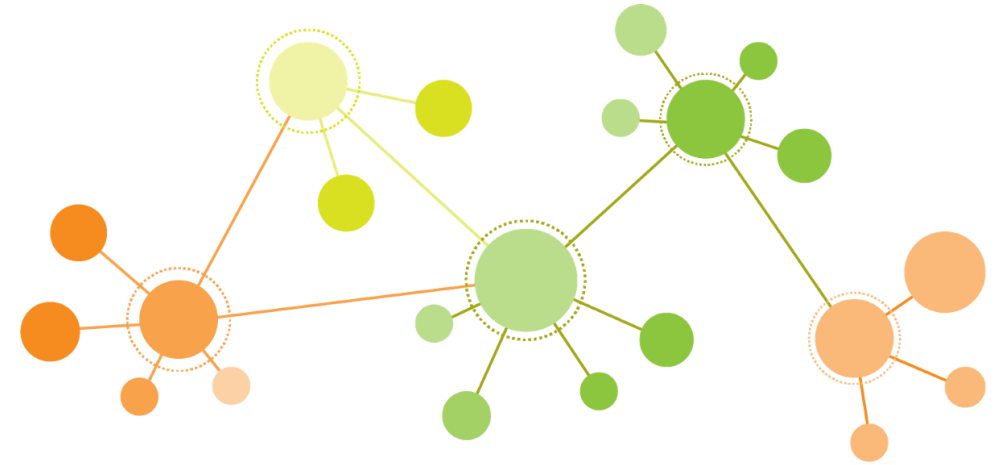
# MANET routing

- **Topológia alapú protokollok**

- Topológia felderítés
  - Ki kinek a szomszédja, milyen költségeik vannak a kapcsolatoknak
- Útvonalválasztás a topológia alapján

- **Helyzet alapú protokollok**

- Földrajzi információk szükségesek
- Forwarding (next hop kiválasztása) a pozíció alapján



# MANET routing

- Topológia alapú útválasztás
  - **Proaktív**
    - Folyamatosan karban tartja a routing táblát, forgalomtól függetlenül
    - Viszonylag stabil hálózatok
    - DSDV – Bellman-Ford algoritmus alapján
  - **On demand, reaktív**
    - Csak akkor épít utat, ha csomagot kell küldeni a célnak, Ideiglenesen tart fent útvonalat
    - AODV
  - Hibrid protokollok
    - Kombinálja az előző kettőt



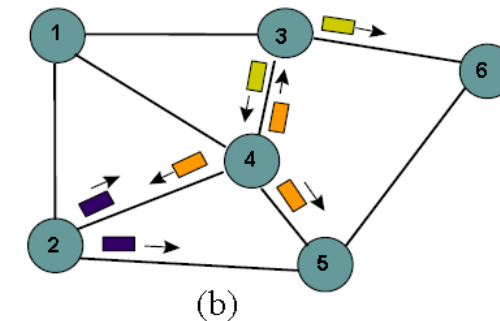
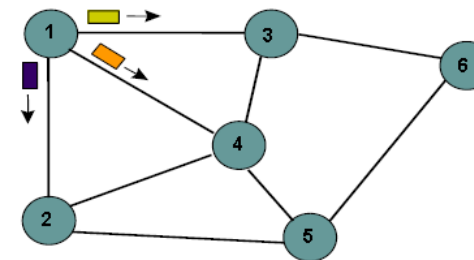
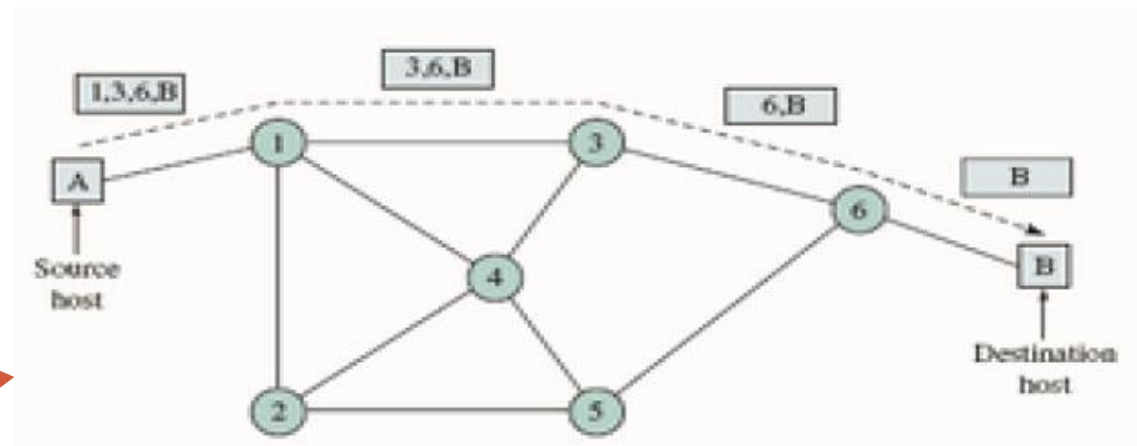
# Megkötések

- Késleltetés
  - **Proaktív** protokollok kisebb késleltetési paramétert biztosítanak, hisz mindig van naprakész útvonal-táblájuk
  - **Reaktív** protokollok ellenben nagyobb késleltetéssel bírnak, hisz meg kell keresni az A-ból B-be vezető utat, ha szükséges.
- Overhead (jelzésforgalom)
  - **Proaktív** protokollok ellenben általában nagyobb overhead-et produkálnak
  - **Reaktív** protokollok esetén kisebb az overhead, hisz a felesleges utakat nem frissítik állandóan
- Az alkalmazásoknál ki kell választani a legmegfelelőbb protokollt
  - Kis mobilitás -> **Proaktív** protokoll
  - Nagy mobilitás -> **Reaktív** protokoll

# Merre küldjem a csomagot?

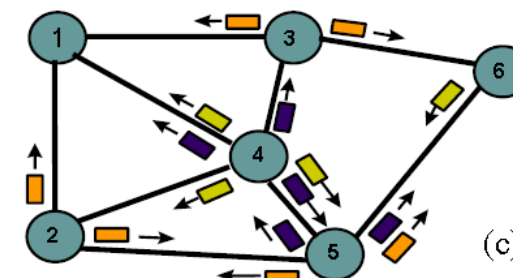
## ▪ Néha csak a forrás tudja

- A fejléc el kell tárolja a köztes csomópontok címét
  - Az útvonal ezt a listát követi
- Forrás-alapú routingnak (**source routing**) nevezik, mert a teljes útvonalat a forrás határozza meg
  - **Pl.: Dynamic Source Routing (DSR)**
- Nagyméretűre nőhet a fejléc
  - Fragmentáció, hatékonyság csökken
  - Különösen hosszabb útvonalak és kis hasznos adat esetén



## ▪ Néha senki se tudja

- Elárasztás alapú (**flooding**) megoldások
  - Mindenki továbbárasztja a kapott csomagot
  - Remélhetőleg eljut a célhoz is
- Nagy a terhelés a vezeték nélküli hálózaton, ahol kevés az erőforrás



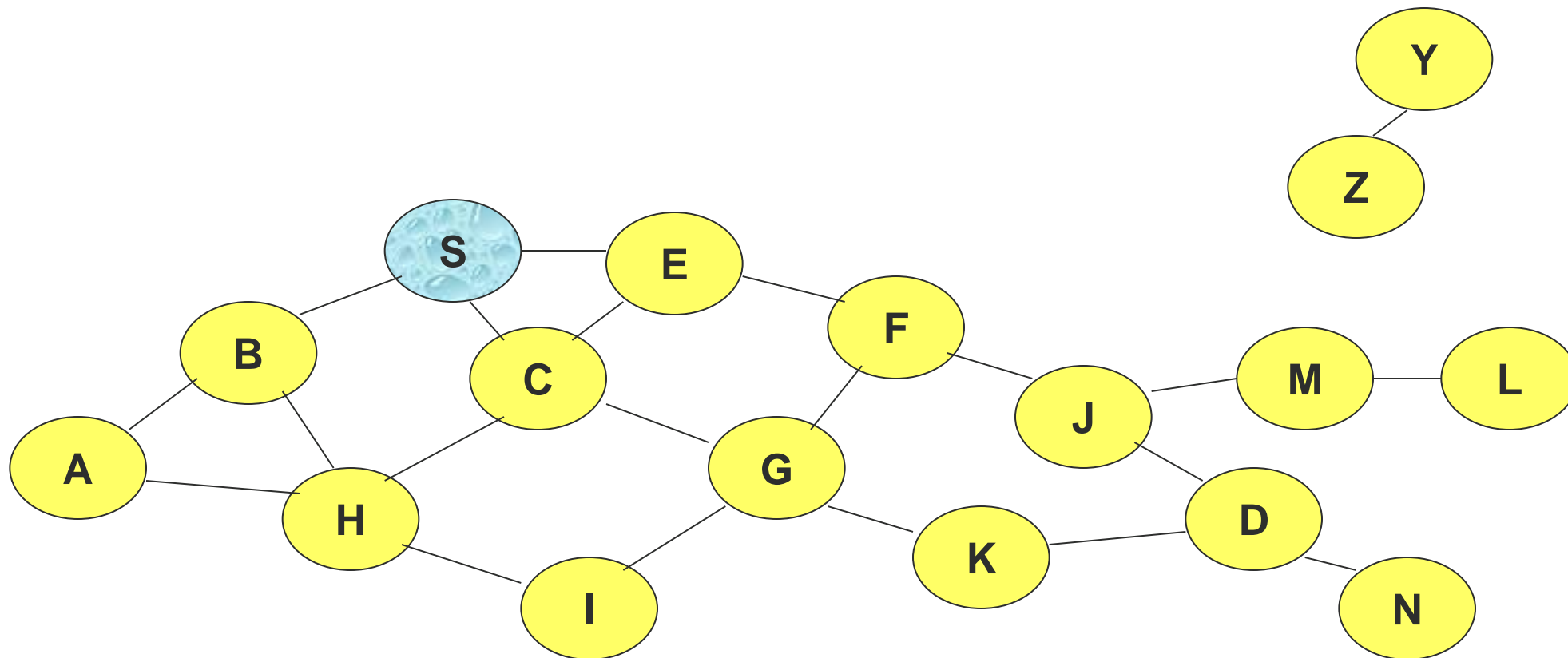
# Ad Hoc On-Demand Distance Vector Routing (AODV)

- AODV igyekszik javítani a helyzetet
  - Routing táblát tart fent, nem kell a fejlécben tárolni a teljes útvonalat
- Ugyanakkor az AODV egy reaktív útvonal
  - Csak az aktív útvonalak esetében tart fent bejegyzést a routing táblákban

- Útvonal felderítéshez a forrás **Route Request (RREQ)** üzenettel árasztja el a hálózatot
  - RREQ-t csak egyszer küldünk, a forrás nem küldi újra
- Amikor egy csomópint újraküld egy Route Request üzenetet, egy **reverse path pointer**-t (visszaút mutató) jegyez be a forrás felé
  - AODV szimmetrikus (bi-directional, kétirányú) linket feltételez
  - Egy rövid időzítő biztosítja, hogy hamar (néhány RTT-nyi időn belül) kiöregedjenek ezek a bejegyzések
- Ha a RREQ eléri a D célállomást, az egy **Route Reply (RREP)** üzenettel válaszol
- A RREP a reverse path pointerek által meghatározott útvonalon éri el az S forrást



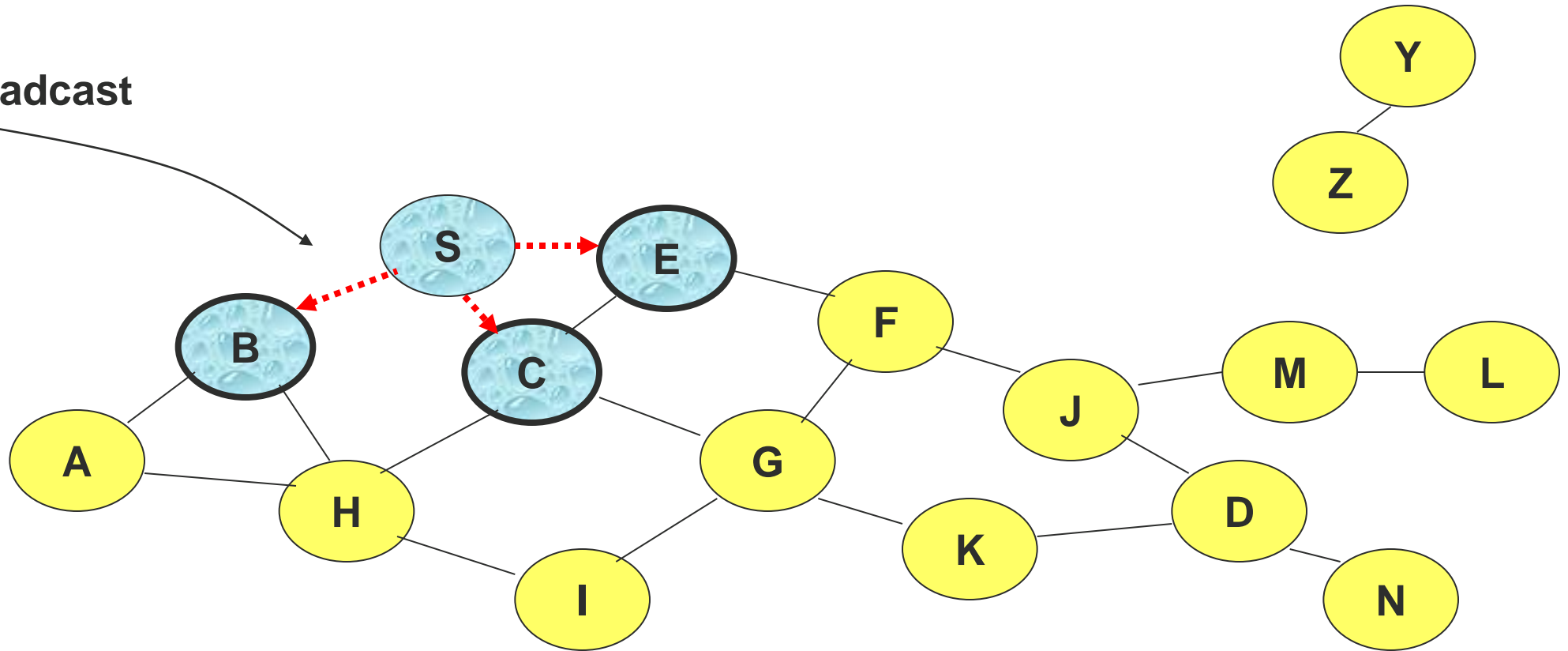
# Route Request - AODV



Olyan csomópont, amelyik már fogadott egy S-ből indított, D-t kereső RREQ üzenetet

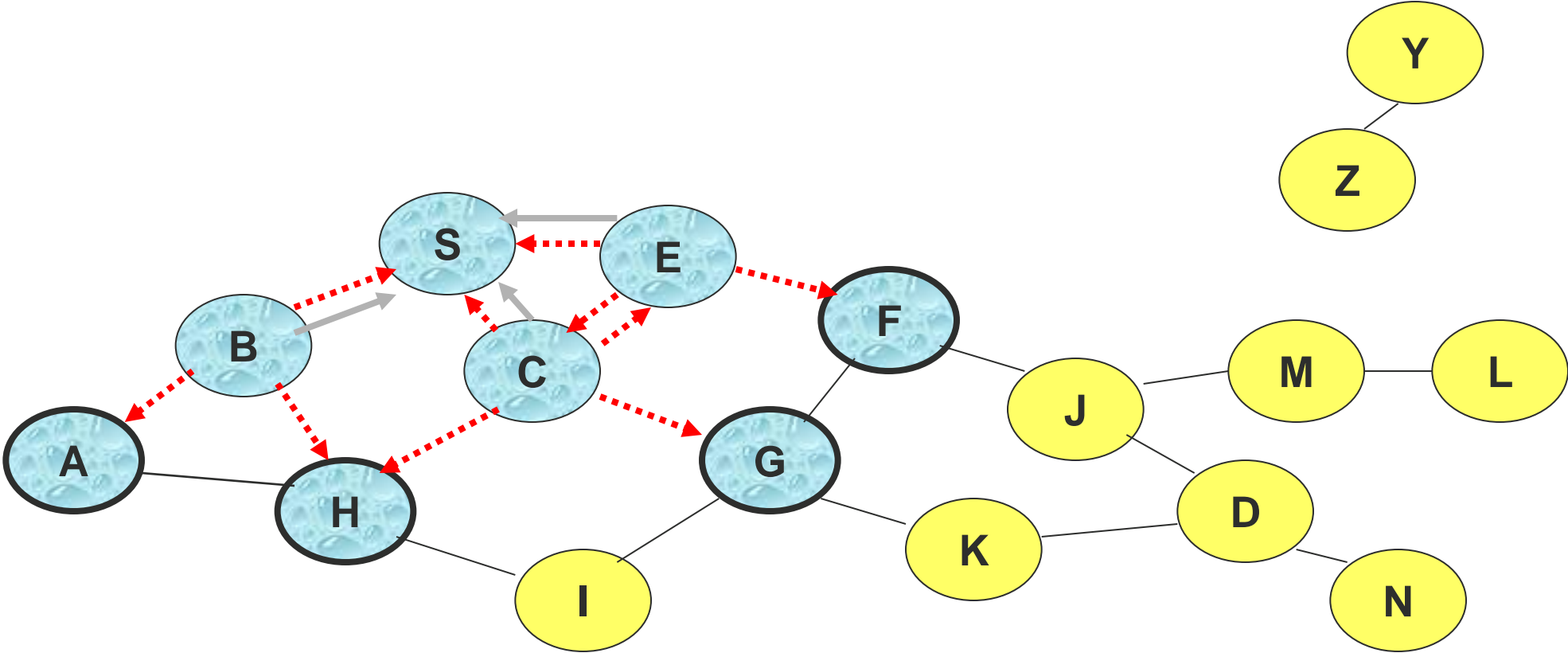
# Route Request - AODV

Broadcast



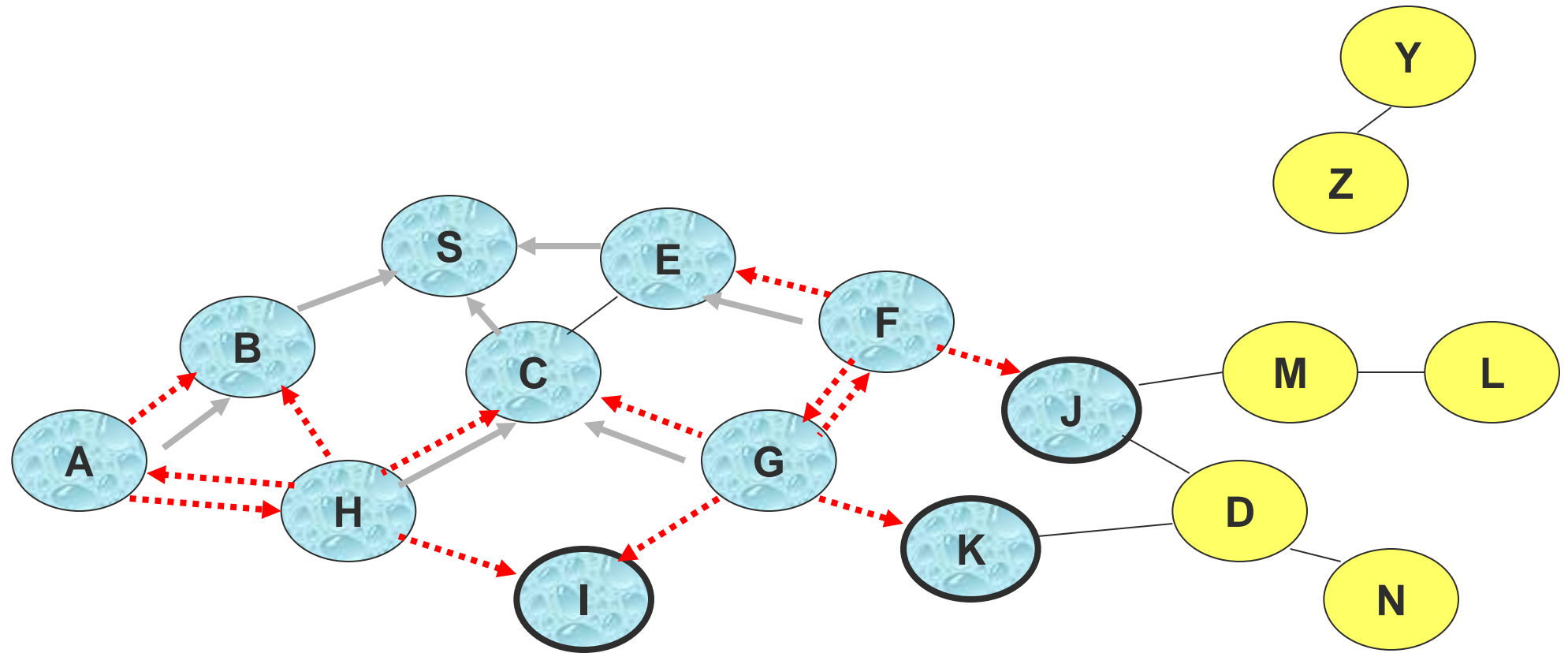
.....➔ RREQ küldése

# Route Requests - AODV



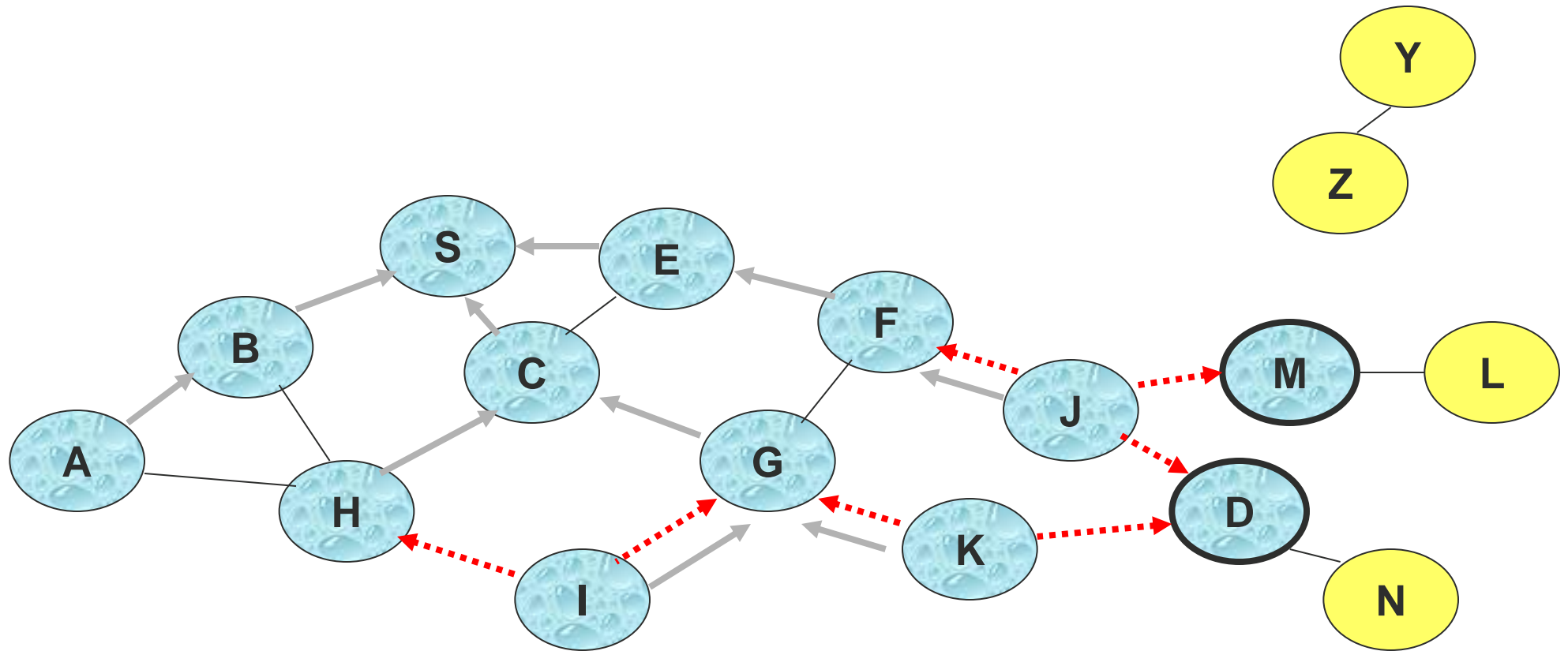
← Reverse Path pointer

# Reverse Path - AODV

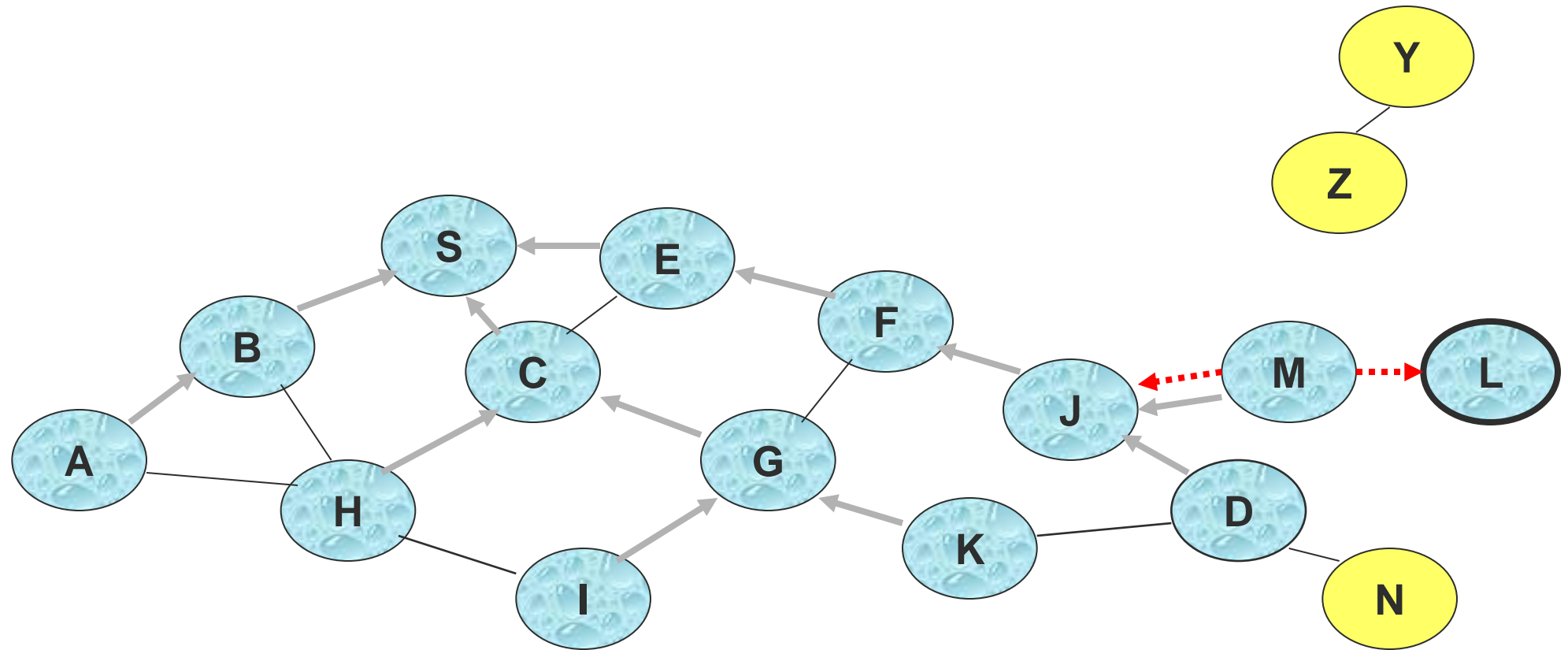


- C csomópont megint kap egy RREQ üzenetet (G és H szomszédjaitól) de **nem továbbítja azt újra**, mivel elküldte azt már korábban

# Reverse Path - AODV

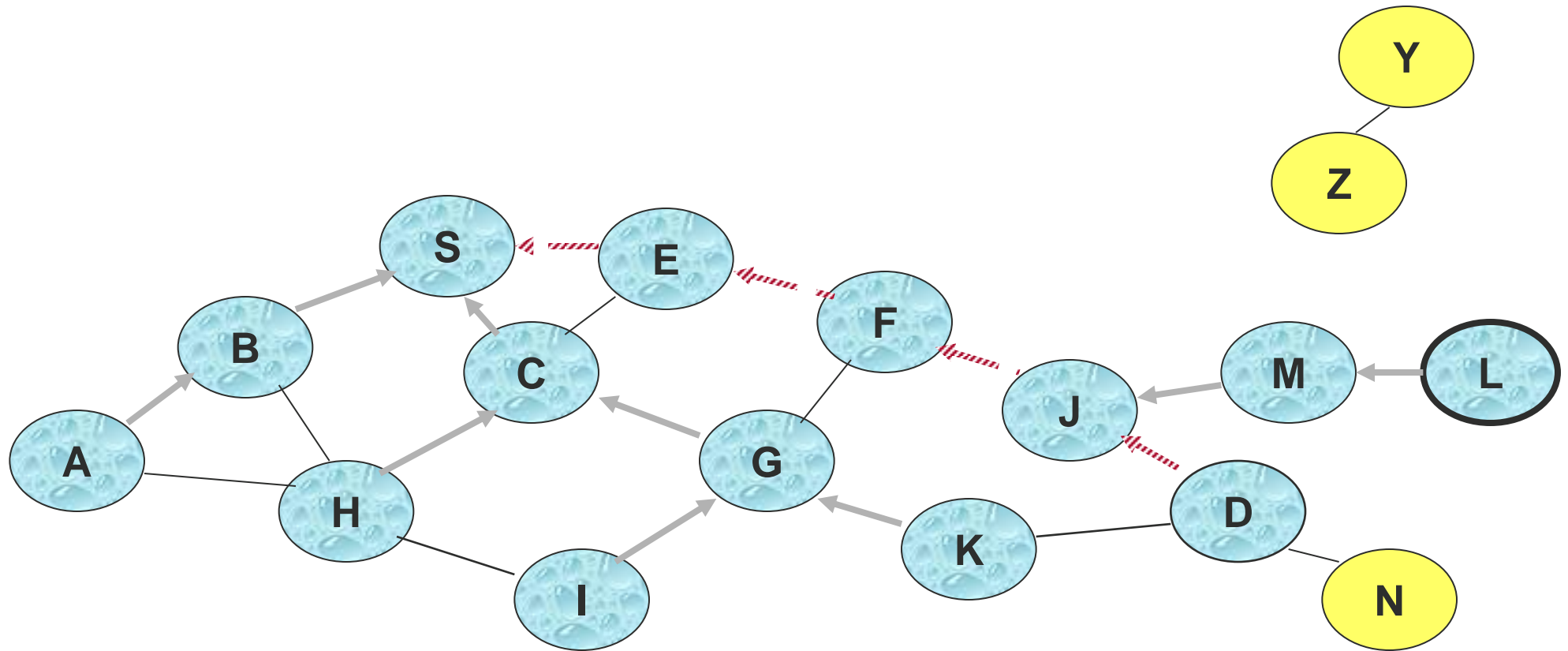


# Reverse Path - AODV



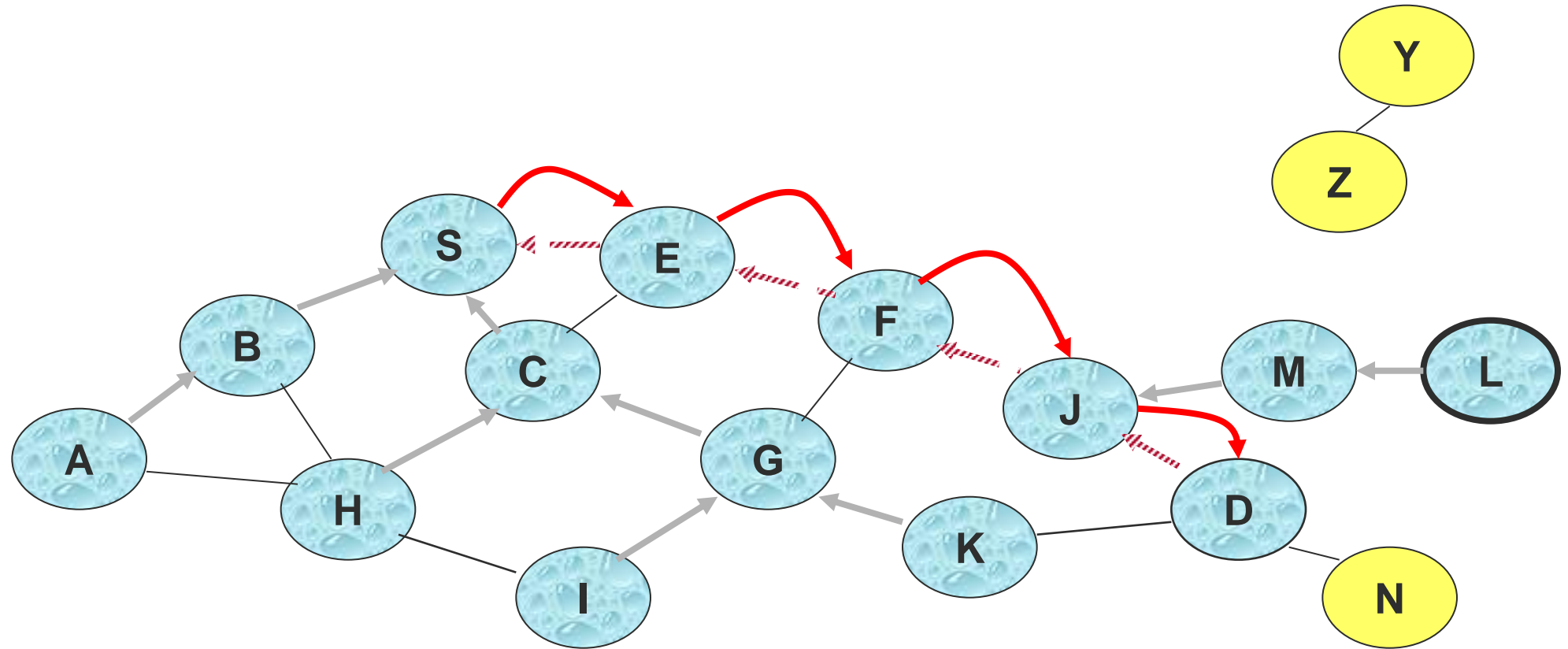
- D csomópont **már nem továbbítja** a RREQ üzenetet, mert ő a célállomás

# Route Reply - AODV



 A RREP üzenet útvonala

# Forward Path - AODV



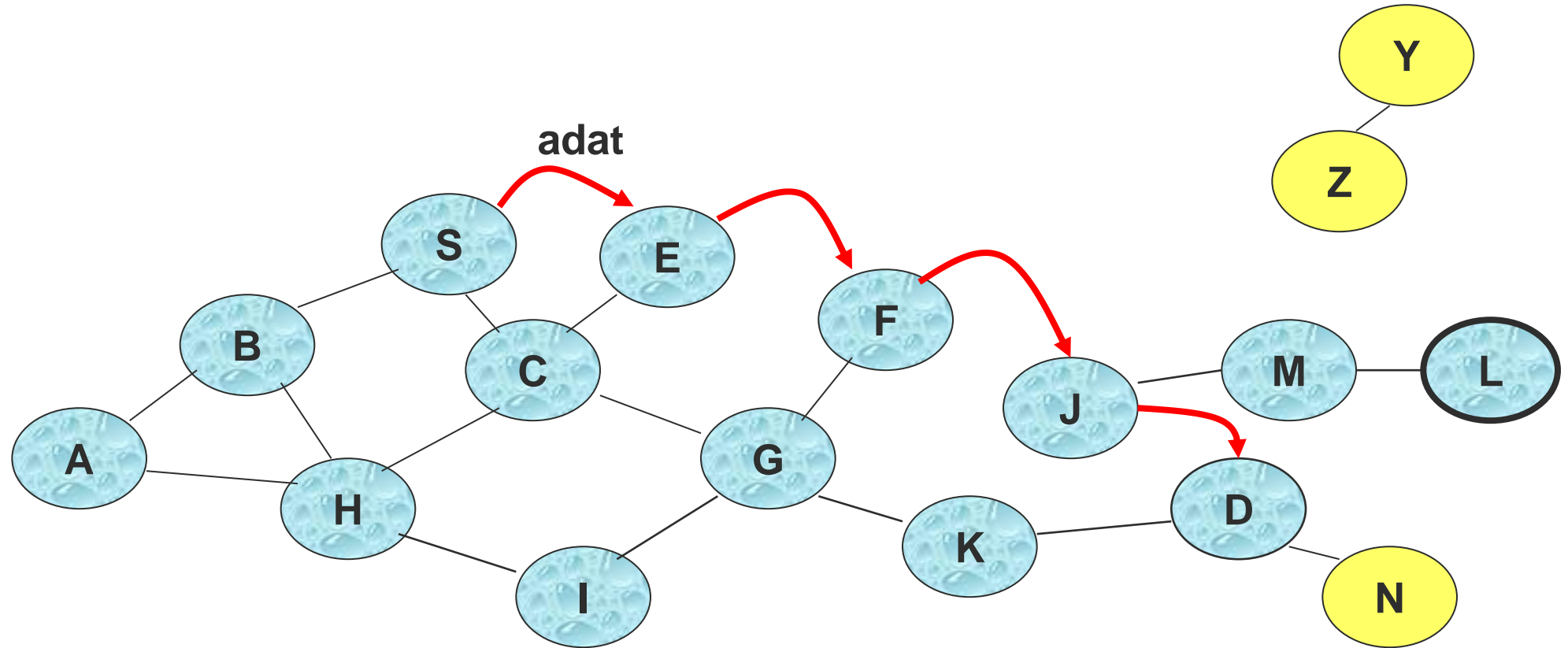
Amint a RREP üzenet a D-től az S-ig terjed,  
bejegyzésre kerülnek az előremutató pointerek



A forward path (előremutató) pointer



# Adatküldés - AODV



Adattovábbításhoz a routing tábla bejegyzéseit használják

Az útvonal **nincs** a fejlécbe foglalva.

# Időzítők

- A **reverse path** bejegyzéseket egy rövid idő után törlik a routing táblából
  - A vezeték nélküli közeg sajátosságait és a hálózat méretét figyelembe véve időt kell hagyni, hogy az RREP üzenet visszaérkezéséig érvényes maradjon a bejegyzés
- Az **előremutató útvonal** bejegyzést akkor törlik, ha inaktívak
  - *active\_route\_timeout*
  - Ha nincs adatforgalom, akkor törlik a bejegyzést, akkor is, ha topológia szerint még érvényes lenne az útvonal

# Optimizálás: Expanding Ring Search

- Növekvő átmérőjű területen történő keresés
- A RREQ üzeneteket először egy kis Time-to-Live (TTL) értékkel küldik
  - Csak néhány ugrásig terjednek
  - Ez használható más, elárasztást használó protokollban is
- Ha nem érkezik Route Reply üzenet, növeli a TTL mező értékét
  - Néhány lépésben a teljes hálózatban keres

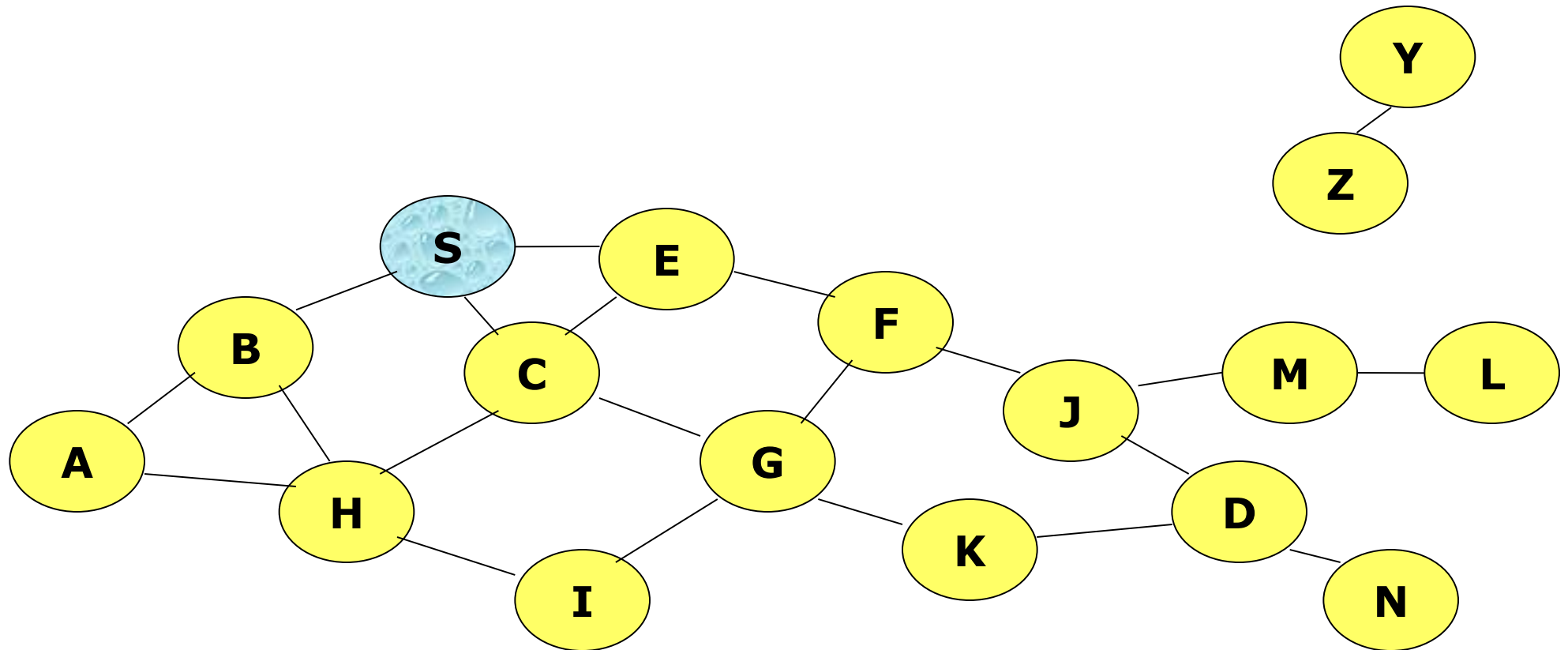
# Összefoglaló: AODV

- Az útvonalak nem a csomagfejlécben vannak meghatározva
- Csak aktív útvonalakra van érvényes routing tábla bejegyzés
- Adott csomópontban egy célállomáshoz legfeljebb egy bejegyzés tartozik
- Használaton kívüli útvonalakat törlik, akkor is, ha a topológia nem változik

# Dynamic Source Routing (DSR)

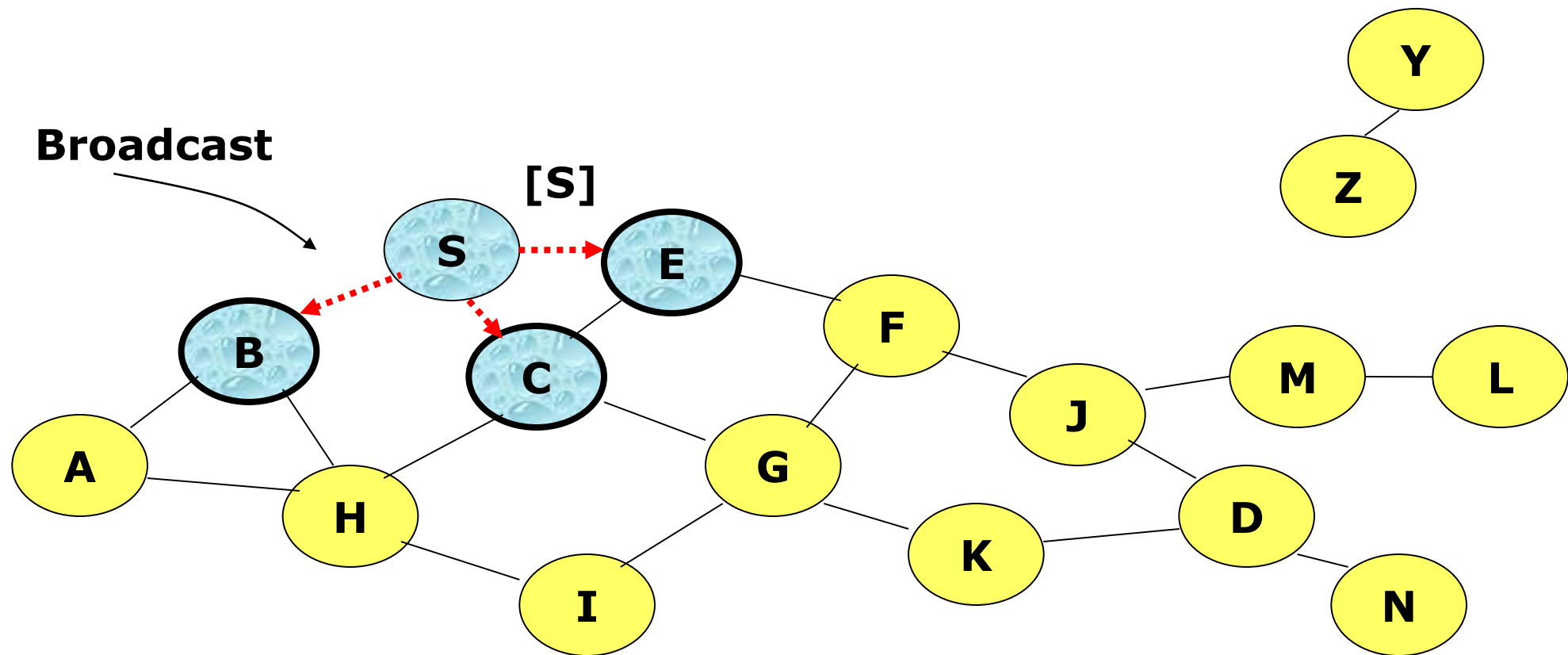
- Ha az S forrás akar valamit küldeni a D célnak, először elindít egy útvonal felderítést
- **S** elárasztja a hálózatot Route Request (**RREQ**) üzenetekkel
- Minden közbeeső csomópont hozzáadja a saját ID-jét a RREQ-hez mielőtt továbbküldené

# DSR Route Discovery



Csomópontok melyek már megkapták a RREQ üzenetet S-től

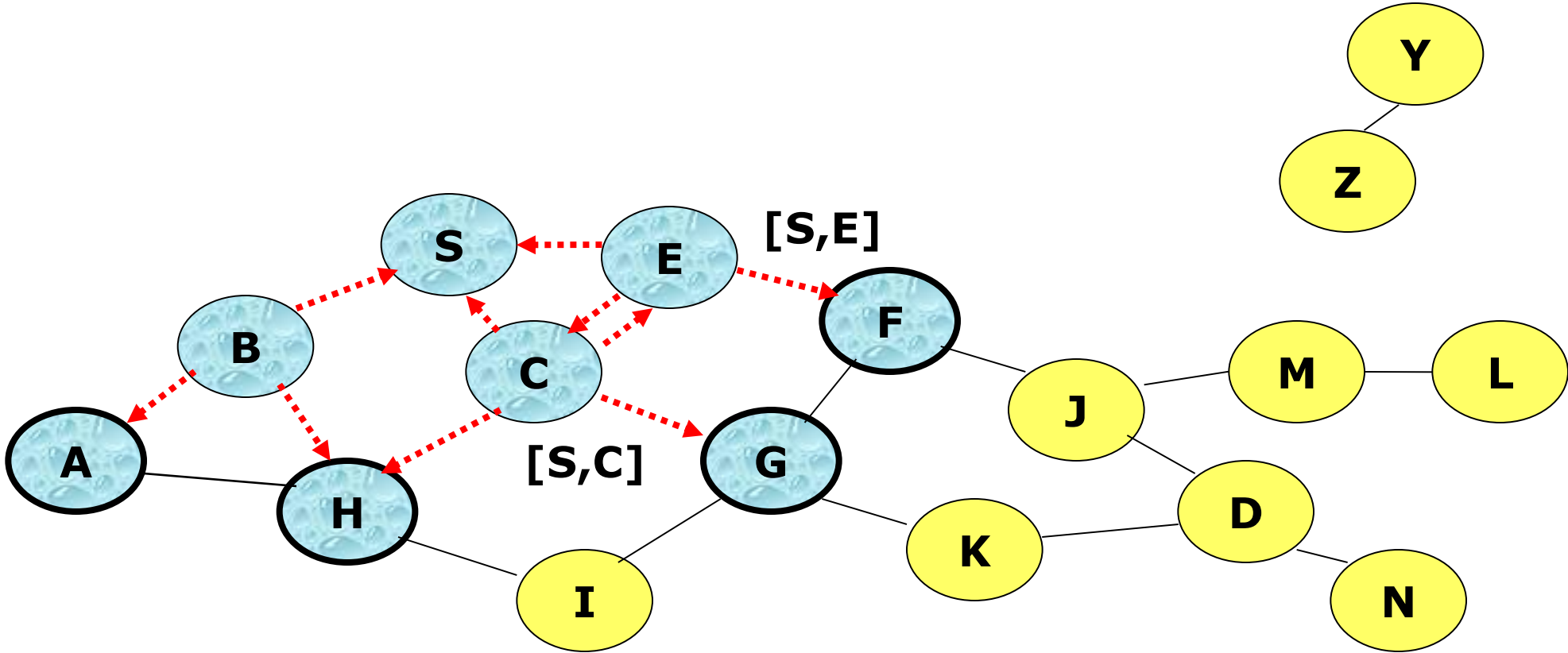
# DSR Route Discovery



**RREQ**

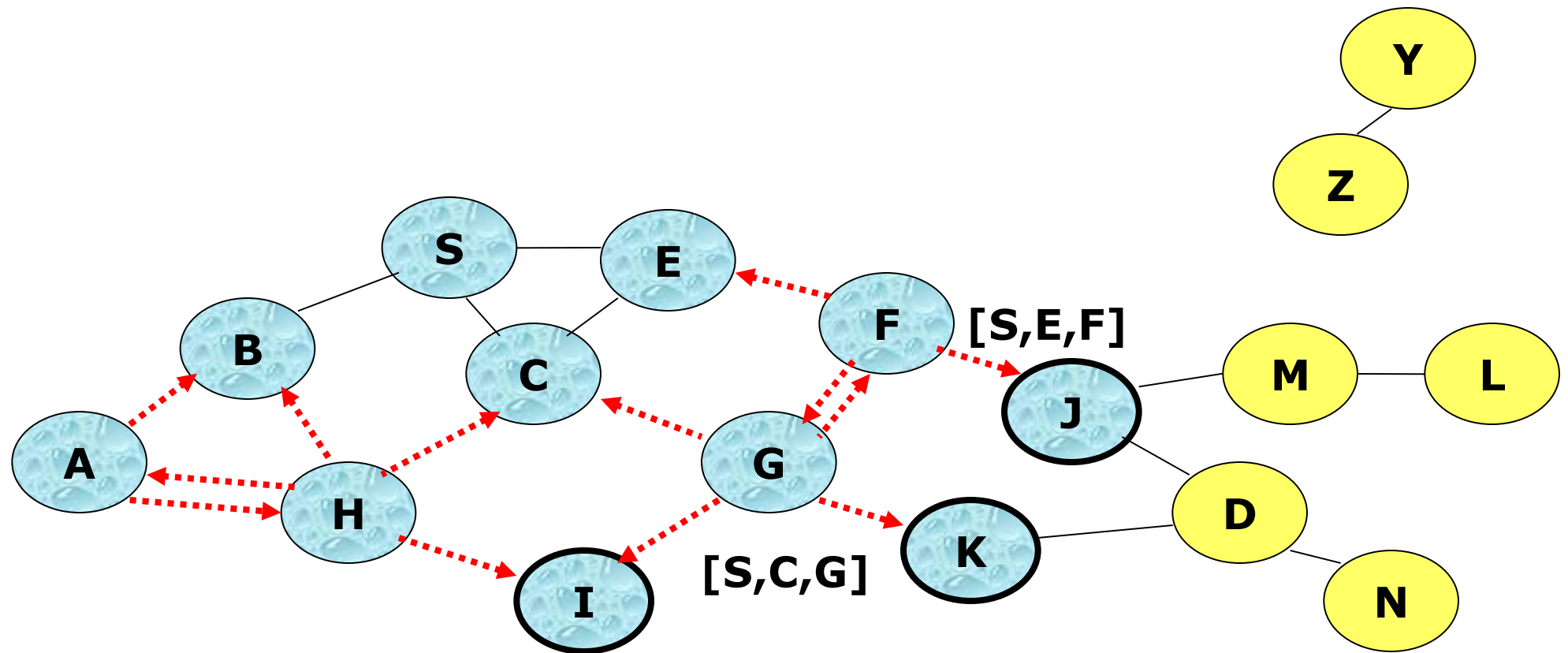
**[X, ...] közbenső csomópontok melyeket már hozzáadtak az útvonalhoz**

# DSR Route Discovery



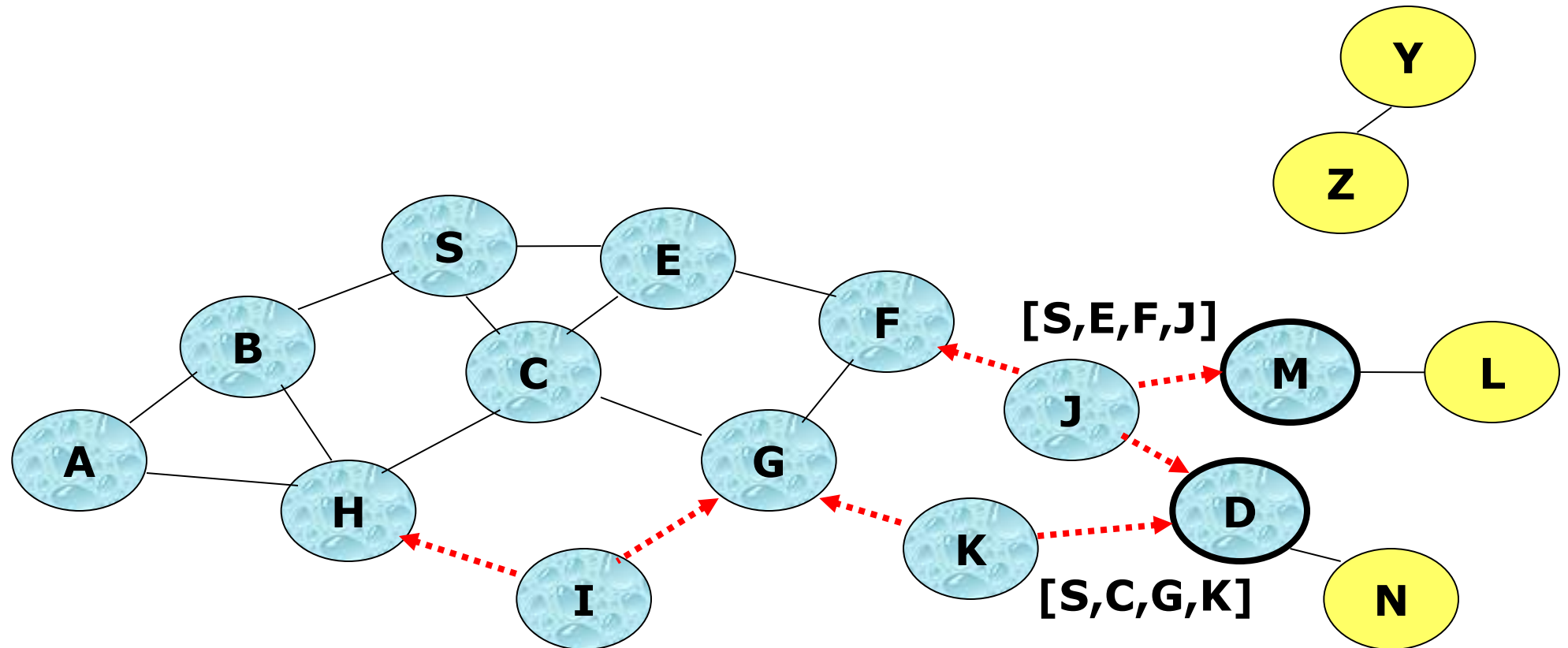


# DSR Route Discovery



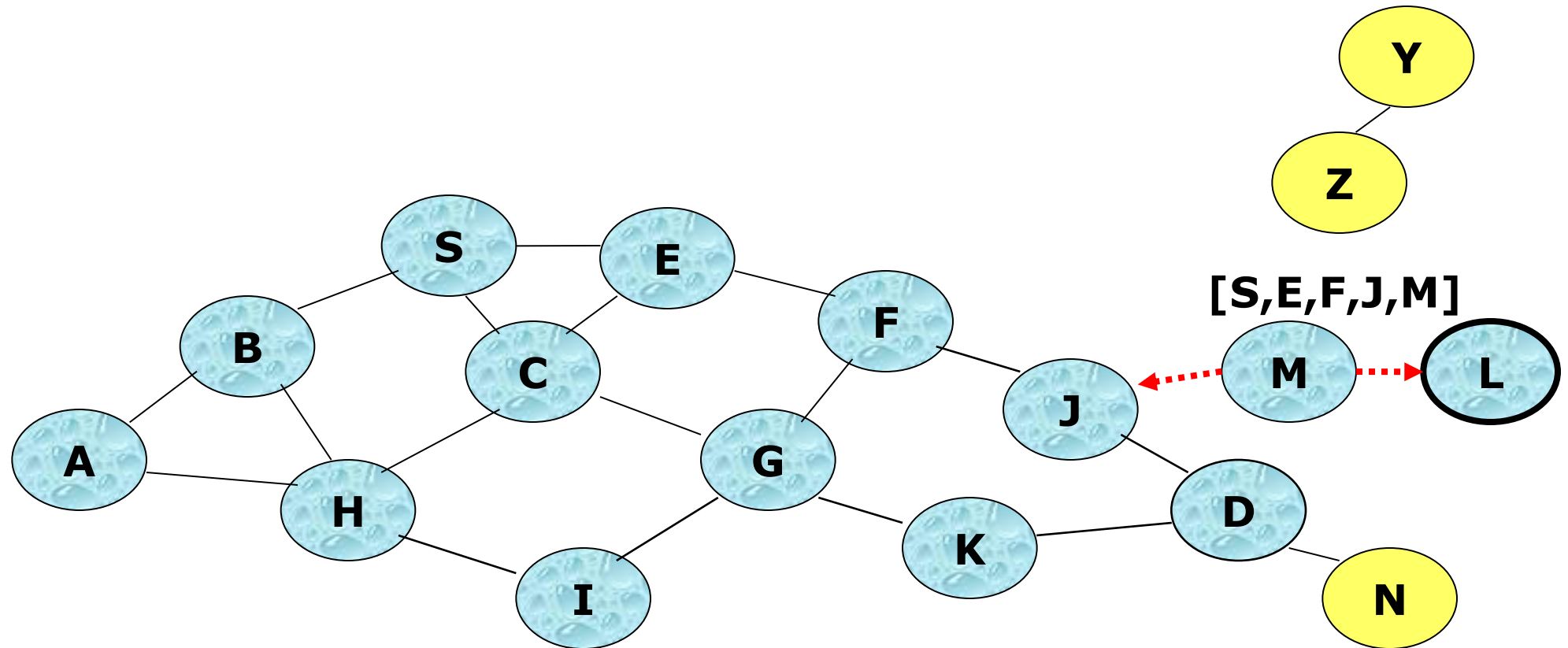
**Korlátozott elárasztás (mint az AODV-ben):  
C ismét megkapja a RREQ-et G-től és H-tól, de nem küldi ismét tovább**

# DSR Route Discovery



**D megkapja a RREQ-et K-tól és J-től, a két csomag ütközhet**

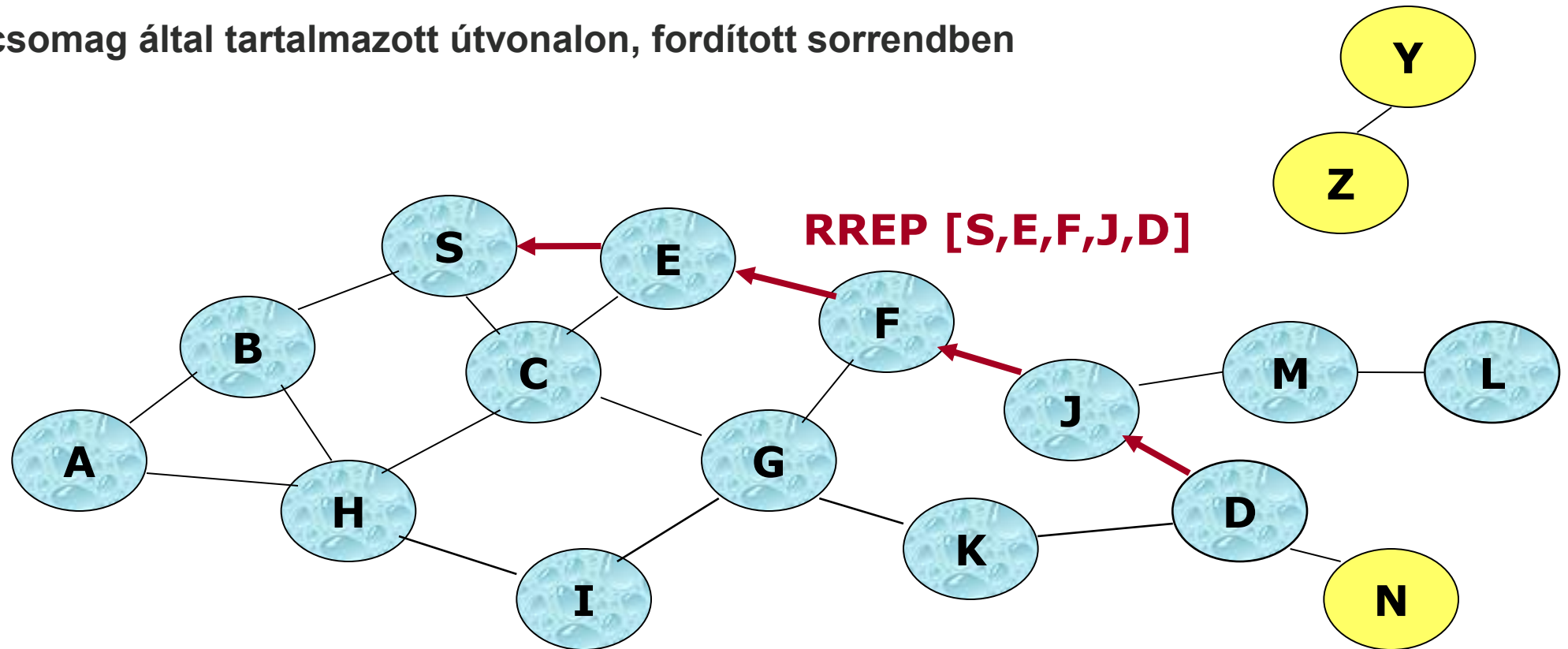
# DSR Route Discovery



**D leállítja az RREQ elárasztás, mivel ő a célállomás**

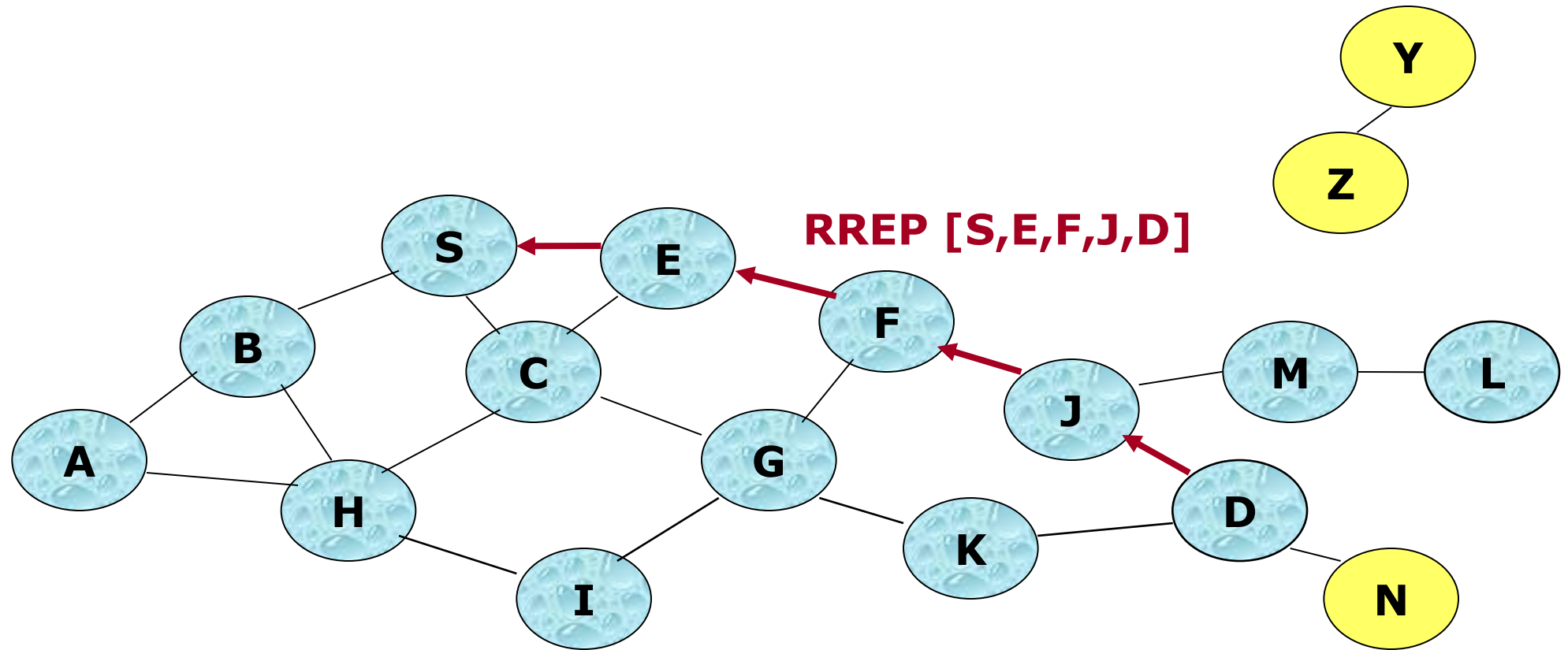
# DSR Route Discovery

- Miután az első RREQ megérkezett, a D célállomás visszaküld egy Route Reply (RREP) üzenetet
- Az RREQ csomag által tartalmazott útvonalon, fordított sorrendben



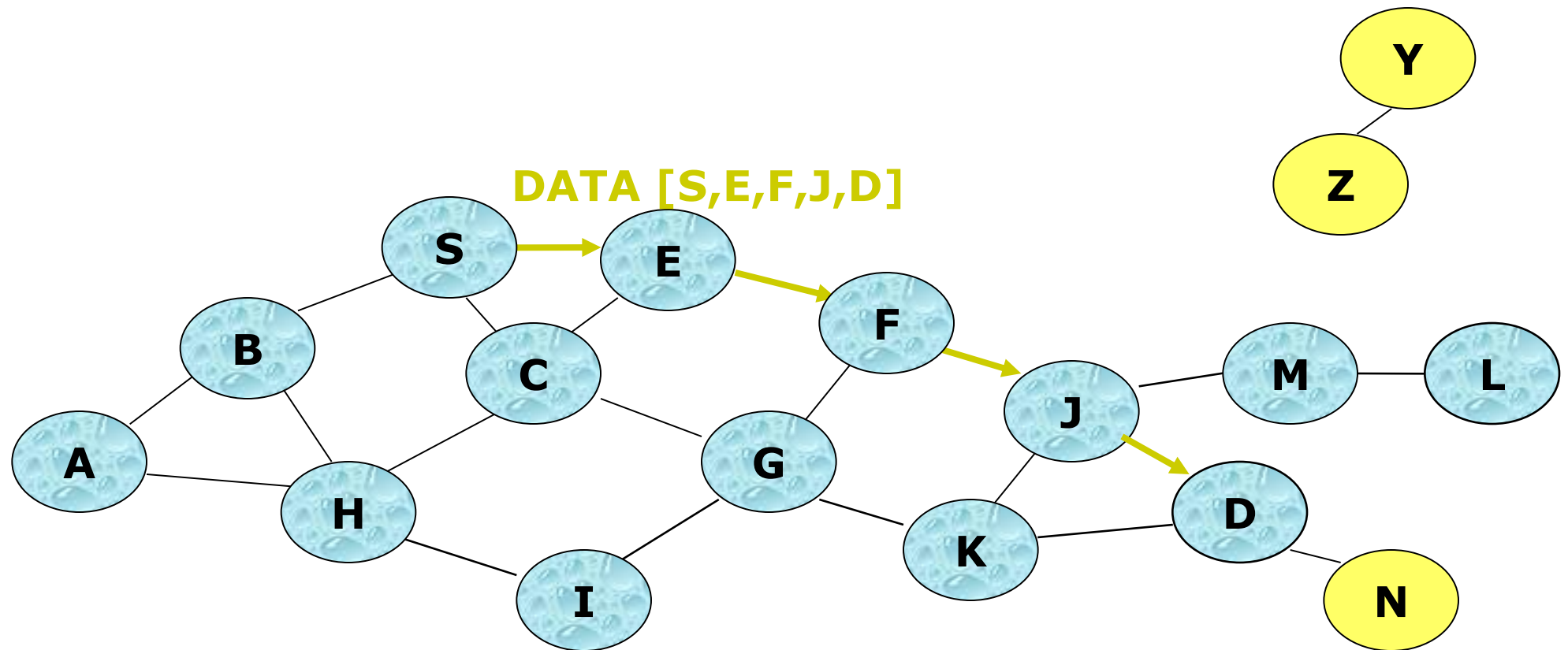
← RREP

# DSR Route Reply



← RREP üzenet

# Adatküldés a DSR-ben



**Az adatcsomagok fejléce nő az útvonal hosszával**

# Pozíció-alapú útvonalválasztás

Ad hoc útvonalválasztás lehet:

- **topológia-alapú** (pl: AODV, DSR)
  - A node-ok közötti összeköttetésekről használ infót
  - Mozgó csomópontok esetén változhat a topológia, de ha nem túl nagy a mozgási sebesség ez nem olyan gyakori
- **pozíció-alapú** (pl: mohó algorimuszok, LAR, DREAM, stb.)
  - **A nodeok fizikai elhelyezkedésének az információját használja fel**
  - Ez lehet: GPS, rádió jelerősség-alapú helymeghatározás, stb.

# Pozíció-alapú útvonalválasztás

- Valamilyen **lokalisációs szolgáltatás** segítségével a küldő *meghatározza a célállomás pozícióját*
  - **Lokalisációs szerver** – ha elérhető akarsz lenni, folyamatosan küldesz friss pozíció adatokat a szervernek
- A cél pozíciójának ismeretében **nincs** szükség *útvonalak felállítására és karbantartására.*
  - Útvonalak helyett: **továbbítási stratégia** - Minden köztes node-nál a cél helyzete alapján történik a következő állomás kijelölése.



# Lokalizációs szolgáltatások

## ▪ Lokalizációs szolgáltatás

- Segít egy adott állomás pozíciójának meghatározásában.
- Ad hoc hálózatban nem mindig áll rendelkezésre egy lokalizációs szerver!

## ▪ Lokalizációs szolgáltatást egy vagy több állomás is nyújthat:

- „néhány/mind-néhánynak/mindnek”

## ▪ Egy küldő – ha nem ismeri a cél helyét – ezeket a lokalizációs szolgáltatásokat veheti igénybe.

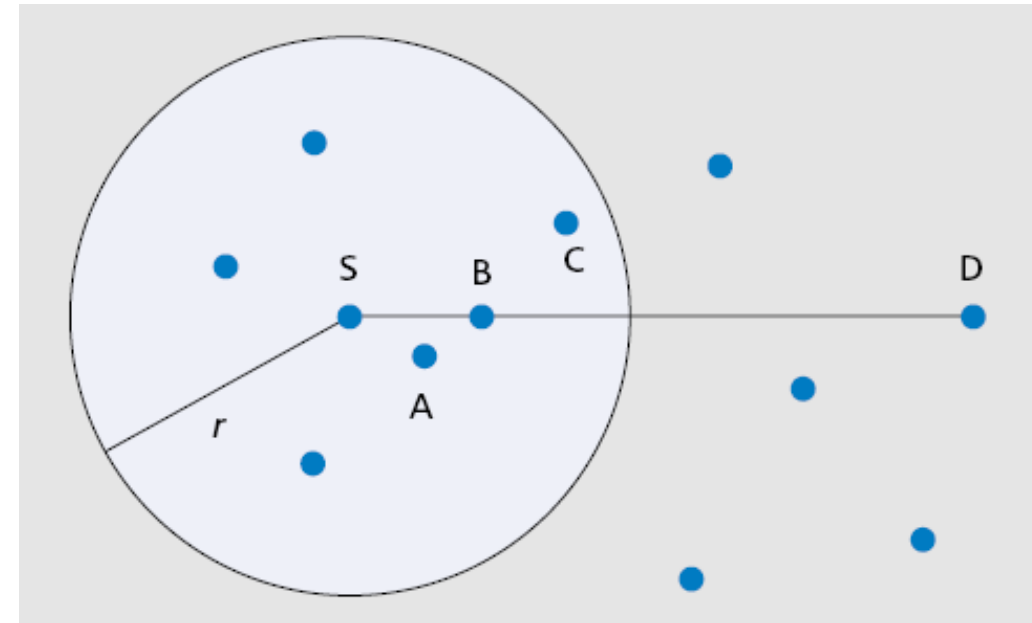
- Pl. cellás (mobil) hálózatban a lokalizáció cella szintű és **központi**.
- Ad hoc rendszerekben ez viszont nem alkalmazható.

# Továbbküldési stratégiák

- Egy köztes node továbbküldési döntése:
  - A csomagban elhelyezett cél pozíciójától függően
  - Az egy ugrásra lévő szomszéd csomópontok pozícióit ismerve történik
- Szomszédok pozíciói: Hello, broadcast üzenetekből, általában periodikusak
  
- Továbbküldési stratégiák:
  - Mohó továbbküldés
    - pl.: MFR, NFP, compass routing
  - Korlátozott célirányú elárasztás (restricted directional flooding)
    - pl.: LAR, DREAM
  - Hierarchikus megoldások

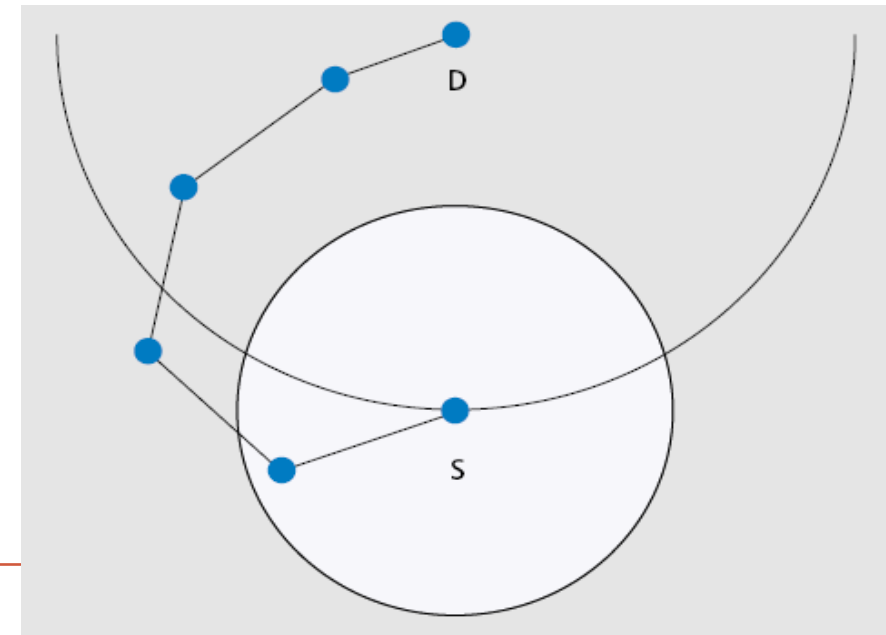
# Mohó továbbküldés

- Milyen stratégia alapján válasszuk ki a köztes node-ot?
- **Most forward within  $r$  (MFR)**
  - Azt a node-ot válasszuk, amely a legközelebb van **D**-hez. (Pl. **C** node)
  - Az ugrások száma ezáltal minimalizálódik
  - Jó stratégia, ha a jelerősséget nem lehet változtatni
- **Nearest with forward progress (NFP)** (pl. **A** node)
  - Ha a jelerősséget lehet változtatni
  - Lecsökken az ütközések valószínűsége
- **Compass routing** (pl. **B** node)
  - Legkisebb szög az SD egyeneshez képest
- **Random D-közeli szomszéd választás**
  - Nem szükséges pontos pozíció infó a szomszédokról
  - Kisebb overhead



# Mohó továbbküldés

- Problémák:
  - **S** közelebb van a **D**-hez mint a többi node
  - Lokális maximumba fullad a küldés, elvileg nincs kiút
  - Gyógyuló (**recovery**) üzemmód:
    - Ha a csomagok küldés lokálisan elakad, akkor átváltunk ebbe a módba.
    - Ha van megfelelő szomszéd a továbbításra akkor pedig visszaváltunk mohó módba



# Location-Aided Routing (LAR)

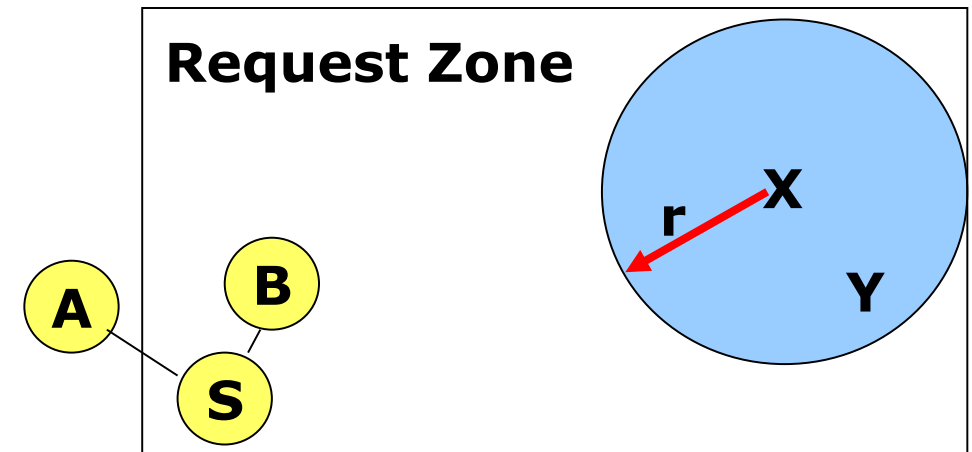
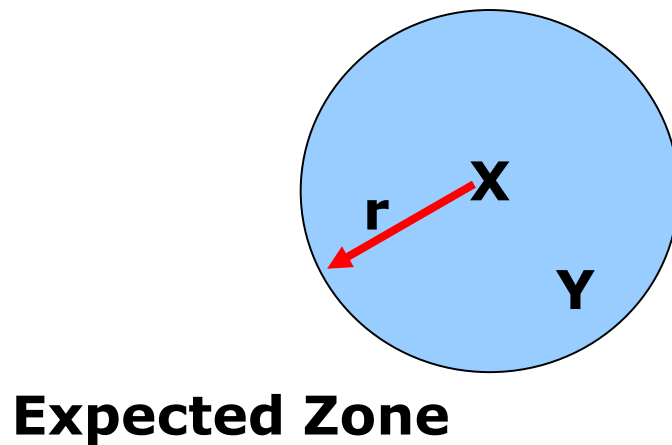
- A célállomás helyzet-információját használja az elárasztás területének *korlátozására*
  - Helyzetinformációt pl. GPS-sel lehet szerezni.
  - Háromszögeléssel, bázisantennákat használva.
- Bevezeti a *várható zóna (Expected Zone)* fogalmát
  - „várható zóna” = az a terület, ahol valószínűleg a célállomás tartózkodik
  - A célállomás korábban ismert tartózkodási helyét és mozgási irányát, sebességét használják fel a becslésre.
- RREQ csak az ún. **Request Zone**-on belül továbbítódik.
  - A Request Zone tartalmazza az Expected Zone-t, illetve a forrástól az Expected Zone-ig húzódó tartományt.

# LAR Expected Zone, Request Zone

**X** = a **D** célállomás utolsó, ismert tartózkodási helye  $t_0$  időpontban.

**Y** = a **D** célállomás jelenlegi, **S** forrás számára ismeretlen tartózkodási helye  $t_1$  időpontban.

$$r = (t_1 - t_0) * [\mathbf{D} \text{ sebességének becsült értéke}]$$



# LAR Request Zone (2)

- Csak a Request Zone-on belüli állomások továbbítják a RREQ-t
  - A Request Zone **lehet például** az Expected Zone-t és a forrást magába foglaló legkisebb téglalap, melynek oldalai párhuzamosak az X és Y tengelyekkel.
  - Pl. az előbbi példán **B** továbbítja a RREQ-t, de **A** nem
- A Request Zone-t explicit módon meghatározza a RREQ üzenet.
- Minden állomásnak ismernie kell saját helyzetét, hogy eldönthesse, beleesik-e a Request Zone-ba.

# LAR Request Zone (3)

- Ha a forrás nem helyesen becsülte meg a célállomás helyzetét, a Request Zone lehet, hogy nem tartalmazza azt  
→ *az útvonalfelderítés nem lesz sikeres!*
- A forrás timeout után új keresést indít, amelynél...
  - növeli a Request Zone területet;
  - szükség esetén az egész hálózatot megjelölve Request Zone-ként.
- A LAR útvonalfelderítésének további lépései megegyeznek a **DSR**-ben leírtakkal
  - Az RREQ üzenetbe lépésről lépésre bejegyezzük az útvonalat
  - A céltól egy RREP üzenet visszajut a forráshoz, benne a teljes útvonallal
  - Ezt az útvonalat beteszi utána a forrás az üzenetek fejlécébe
  - Az útvonalak elavulnak, időnként frissíteni kell



# LAR változatok: Adaptív Request Zone

- Az RREQ-ben tárolt Request Zone-t minden belső állomás módosíthatja, ...
  - amennyiben frissebb/pontosabb információja van a célállomásról,
  - ÉS amennyiben az eredmény egy kisebb Request Zone lesz.



# LAR összefoglalás

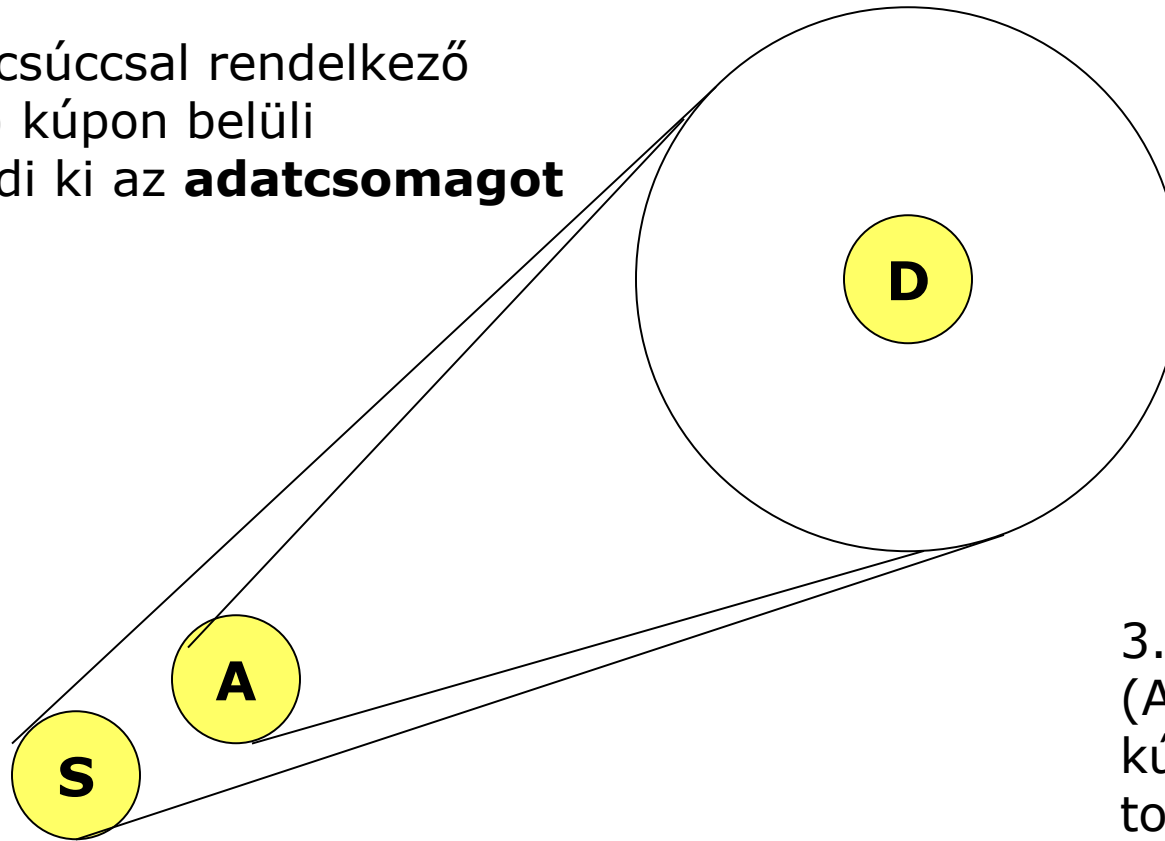
- Előnyök
  - Az RREQ elárasztás területét korlátozza
  - Az útvonal felderítés overhead-et csökkenti
- Hátrányok
  - A csomópontoknak ismerniük kell a fizikai elhelyezkedésüket.
  - Nem veszi figyelembe az esetleges rádiós átvitelt blokkoló akadályokat.

# Distance Routing Effect Algorithm for Mobility (DREAM)

- Helyzeti- és sebesség információt használ (mint a LAR) az adatcsomagok elárasztási területének leszűkítésére.
- Elárasztással terjeszti az **adatcsomagokat** (a LAR-ral ellentétben, ahol útvonalfelderítés van)

# DREAM lokalizálás

2. S forrás az S csúccsal rendelkező (nagyobb) kúpon belüli szomszédjainak küldi ki az **adatcsomagot**

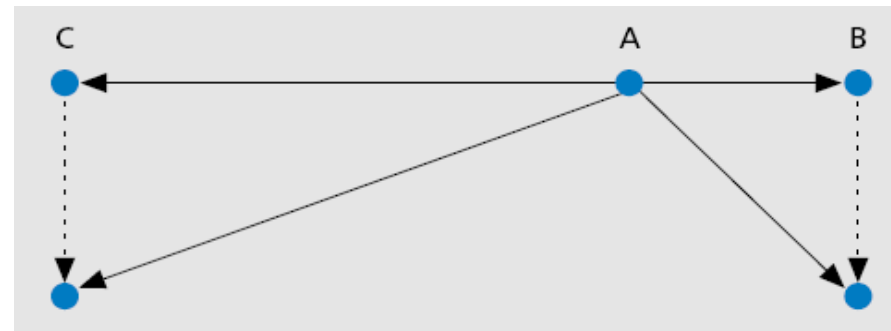


1. Expected zone („LAR értelemben”)

3. Az **A** belső állomás a jelzett (A csúccsal rendelkező) kúpon belüli szomszédjainak továbbítja az **adatcsomagot**

# DREAM távolsági hatás

- Az állomások periodikusan hirdetik (üzenetszórással) saját helyzetüket.
- „Távoli hatás” = A távoli állomások kisebb szögsebességgel, azaz látszólag lassabban mozognak.



- - *A közeli állomásokat gyakrabban kell frissíteni, míg a távolikat elég ritkábban.*
  - Helyzet frissítő üzenetek *time-to-live* (TTL) mezőt használnak a terjedés szabályozására
  - Változó TTL értékekkel lehet a távoli frissítéseket ritkítani

# Versengés-alapú továbbítás (CBF)

- Tipikusan pozíció-alapú routing megoldásokban egy node beacon üzenetektől tudja meg a szomszédok helyzetét.
  - A mobilitás, energia-gazdálkodás sokat ront a helyzeten, a beaconing frekvenciája nagyban változhat.
- Javaslat: **CBF (Contention Based Forwarding)**
  - **mohó küldés szomszédos ismeretek nélkül!**
  - A kiválasztás a csomópontok aktuális helyzete alapján fog megtörténni csomagtovábbítás *közben*.

# Versengés-alapú továbbítás

- CBF elemei:
  - Továbbító kiválasztása: **versengés alapján** történik
  - **Elnyomás**: lecsökkenti az ütközések esélyét, hogy lehetőleg ne választódjék ki egynél több node.
- Lépések:
  1. A küldő a csomagot üzenetszórással elküldi a szomszédoknak.
  2. A szomszédok **versengés alapján** meghatározzák, hogy ki legyen a jogosult a továbbításra.
    - PI. Időzítők értéke fordítottan arányos a küldőhöz képesti távolsággal
  3. A nyertes node **elnyomja** a többi node küldési hajlamát.

# Járművek közti kommunikáció

- „Hagyományos” ad hoc protokollok:
  - Reaktív: AODV, DSR
    - nagy útfelderítési terhelés, lassú felderítés
    - járművek között nem mindig hatékony
  - Geográfiai-alapú: LAR, DREAM
    - Lokalizációs szolgáltatás szükséges
    - bizonyos esetekben viszont lokális maximumba vezetnek (recovery mód)
    - városi környezetben nem hatékony
  
- Új megoldásokra van szükség a járművek közti (V2V) kommunikációban





# AODV verziók VANET-re

## ▪ AOMDV: Multipath

- Nemcsak egy útvonalat jegyez fel, hanem mindet, amit talál
  - Ez megtehető extra körök nélkül, mert a felderítés amúgy is elárasztás alapú
- Ha az elsődleges útvonal megszakad, akkor gyorsan át lehet kapcsolni valamelyik tartalékra
  - Akkor kell újra útvonalat keresni, ha minden ismert útvonal megszakadt (vagy erősen fogynak a lehetőségek)

## ▪ SD-AOMDV: Speed and Direction

- A nodeok figyelembe veszik a sebességüket és mozgási irányukat
- Csak olyan node lehet next hop, amelyik ugyanabba az irányba megy, hasonló sebességgel

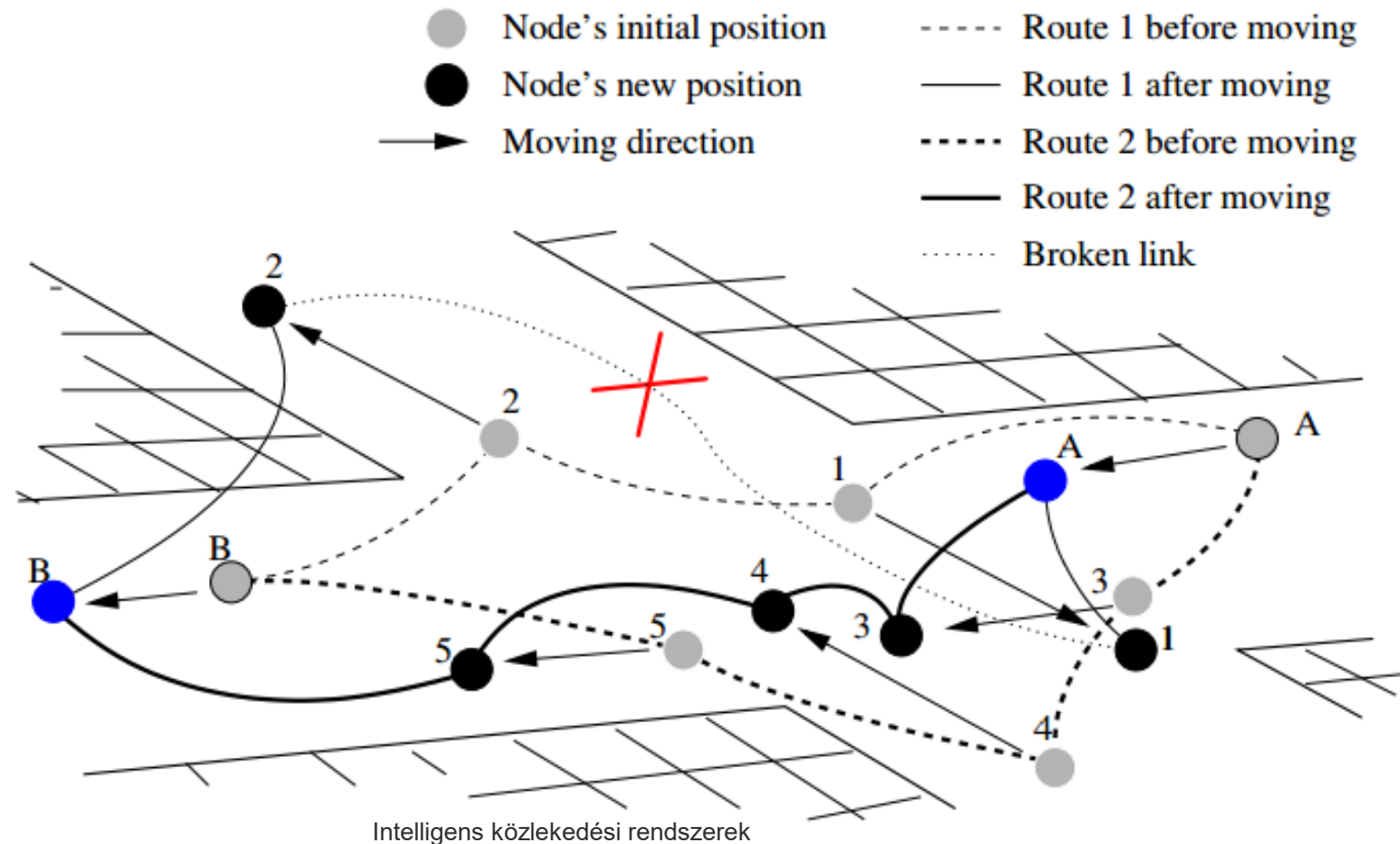
## ▪ R-AOMDV: Retransmission count

- Figyelembe veszi a linkek minőségét a routing metrikában a hopszám mellett
  - Link minősége: MAC újraküldések száma, mielőtt sikeres az átvitel
- **Probléma:** a linkek minősége nagyon gyorsan változik
- Cross-layer optimalizálás

# Link-stabilitás alapú útválasztás

## ▪ Movement Prediction based Routing (MOPR)

- Figyelembe veszi az autók pozícióját, sebességét, irányát
- Olyan közbeeső csomópontokat keres az útvonal kiépítéséhez, melyek „hasonlóan” mozognak

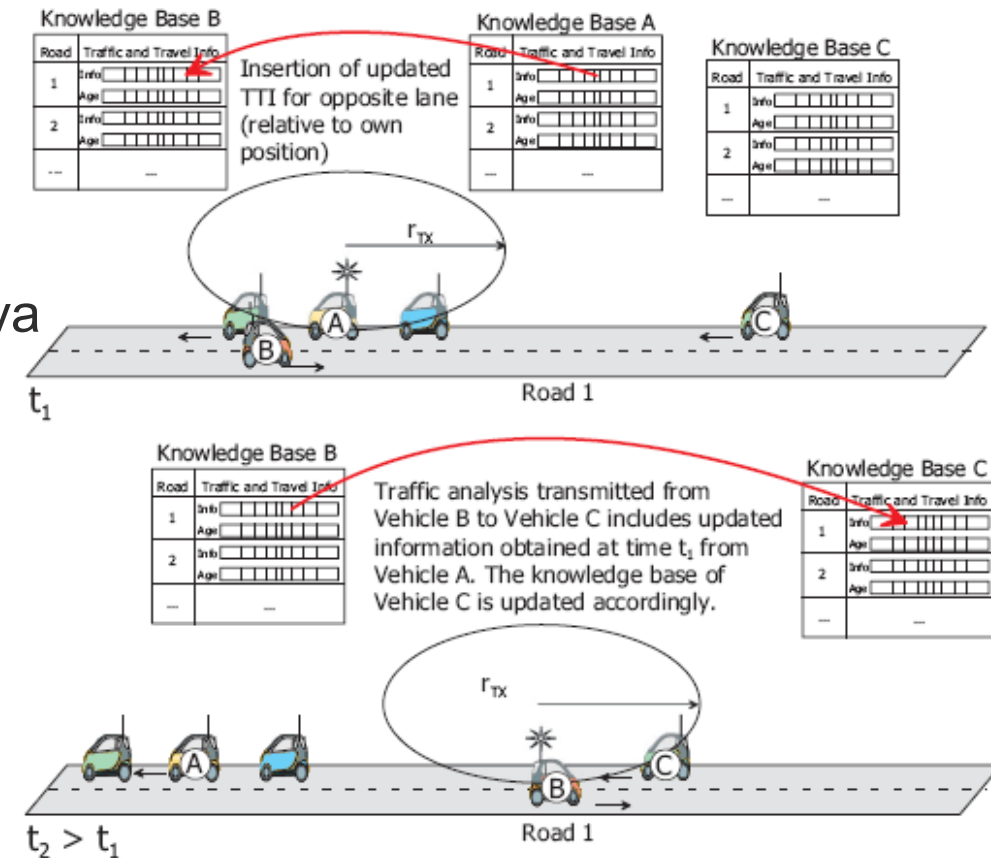


# AODV verziók VANET-re

- **AODV+PGB: Preferred Broadcast Group**
  - A next hop ha túl közel van, akkor nem halad az üzenet
  - A next hop ha túl távol van, akkor könnyen megszakadhat a link
  - **Javaslat:** azokat a szomszédokat választjuk next hop-nak, akiknek közepes jelerősséggel vesszük az adását, ők a PBG
  
- **BAODV: Bus-AODV**
- **P-AODV**
- **I-AODV**
- **Improved-AODV**
- **AODV-BD**
- **AODV-VANET**
- **etc.**

# DTN: Delay Tolerant Network

- Ha ritkán vannak a nodeok, akkor megszakadhat a hálózat
- Ezt **carry-and-forward** módszerrel át lehet hidalni
  - **Data-mule** (adathordozó öszvér)
- Ez akkor lehetséges, ha az üzenet nem veszti el az érvényességét időközben
- Mobilitás predikció nagyon hasznos, ha jól van megvalósítva



# Intelligens elárasztás pletykálással

- Az üzeneteket továbbszörjük/eldobjuk egy bizonyos  $p$  valószínűséggel
  - **Carefully Localized Urban Dissemination (CLOUD)**
- Az eldobás valószínűsége függ attól, hogy egy adott útszakaszon levő autók mekkora valószínűséggel mennek a veszélyforrás felé
- Forgalmi adatbázis szükséges
  - Kanyarodási valószínűség minden útkereszteződésben
  - Megállás valószínűsége minden útszakaszon
  - Átlagos forgalom sűrűség az adott napszakban
- Megbízhatóság növelése egy szavazásos mechanizmussal
  - A csomagot csak akkor dobjuk el, ha megfelelő számú szavazat érkezett erre
- Miklos Mate, Rolland Vida, „Reliable Gossiping in Urban Environments”, in Proceedings of 72nd IEEE Vehicular Technology Conference VTC-Fall, Ottawa, Canada, September 2010.

# Intelligens elárasztás pletykálással

- Szimulációs eredmények a CLoUD protokollra
  - Budapest digitális térképe

