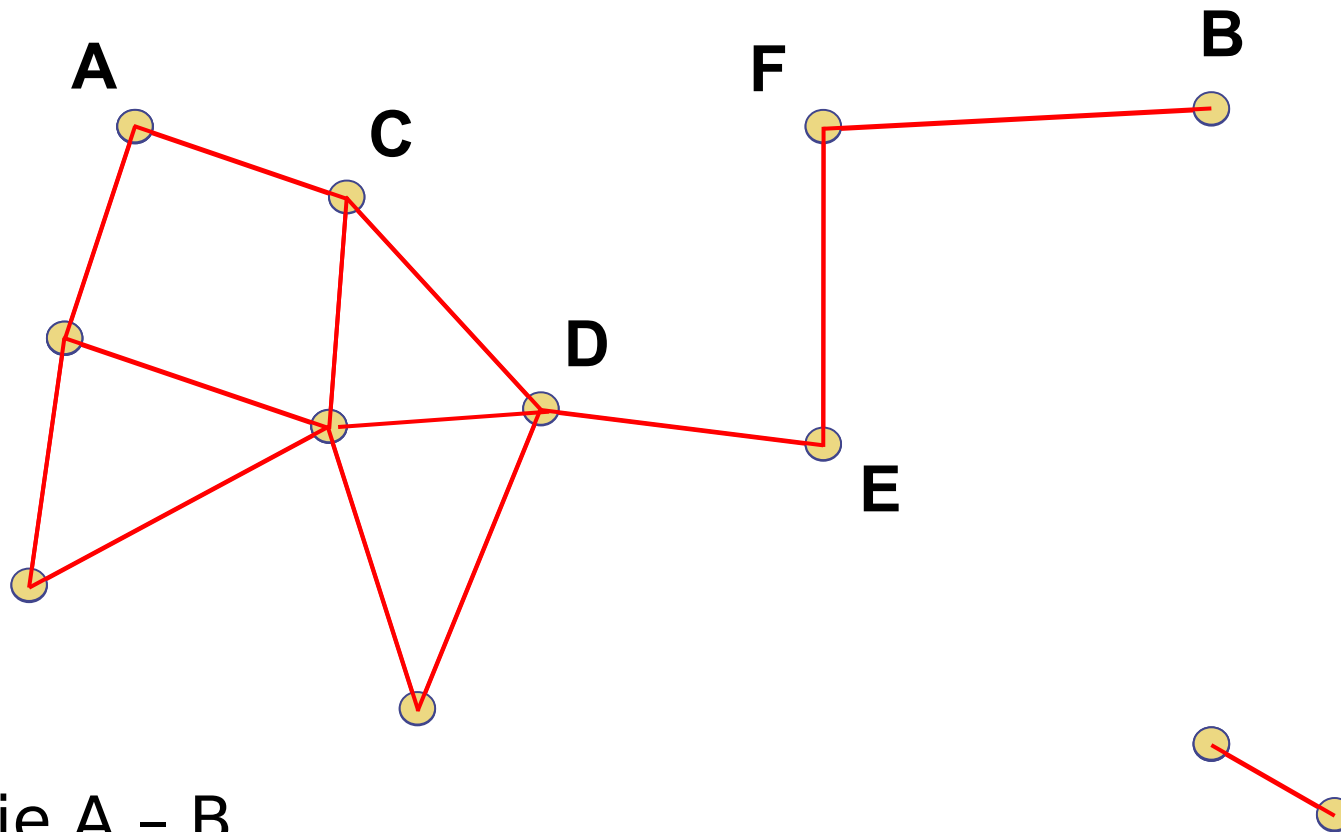


# Počítačové siete 2

## Smerovacie protokoly 1

Martin Drozda

# Ad hoc sieť



Spojenie A - B

Smerovanie paketov cez uzly: A-C-D-E-F-B

# Smerovanie

Pre smerovanie je potrebný výpočet cesty pre dátové pakety, napr. v našom prípade je potrebné nájsť cestu z uzla A do uzla B.

Každý uzol môže mať smerovaciu tabuľku, v ktorej sú zaznamenané už vypočítané cesty. Tieto cesty môžu byť úplné (kompletná cesta z A do B), alebo môže byť v tabuľke len nasledujúci uzol na ceste (uzol C).

V prípade, že cesta nie je známa je potrebný výpočet, ktorý môže byť prevedený preposielaním krátkych paketov v sieti, až pokiaľ takýto paket nie je prijatý cieľovým uzlom.

Ciest môže byť vypočítaných niekoľko, dátové pakety môžu byť preposielané len na jednej ceste, alebo na viacerých.

Smerovanie je na sieťovej vrstve.

# Smerovanie v ad sieťach

Smerovanie v ad hoc sieťach je náročné z dôvodu pohybu uzlov. Po nájdení cesty sa môže zmeniť topológia a cesta je neplatná.

Smerovacie pakety, ktoré sú potrebné na výpočet cesty k cieľovému uzlu, spotrebúvajú energiu, ktorá v prípade zariadení napájaných z baterií môže byť výrazne obmedzená.

Prenos smerovacích paketov vyžaduje prístup k médiu, teda MAC protokol. V prípade hľadania cesty je smerovací paket prijatý každým uzlom v sieti, alebo aspoň uzlami do určitej vzdialenosti od štartovacieho uzla A.

# OSI referenčný model



L7: Aplikačná vrstva (Web, email client, ...)

L6: Prezentačná (reprezentácia dát)

L5: Relačná (spojenie štart/ukončenie)

L4: Transportná (TCP, UDP)

L3: Sieťová (smerovanie) (IP; AODV, DSR, TORA, LAR)

L2: Linková (MAC) (802.11, CSMA, MACA, Aloha)

L1: Fyzická (Hardvér, modulácia)

# QoS pre smerovanie

- Oneskorenie (ms)
- Spravodlivosť (Jainov index)
- Priepustnosť (throughput)
- Užitočná priepustnosť (goodput)
- Overhead (réžia)
- Kvalita nájdených ciest (dĺžka, priepustnosť)

# Model smerovania

$G(V(t), E(t))$  alebo  $G(V(t), E(t), w_1(t), w_2(t))$

$w_1, w_2$  je ohodnotenie hrán a uzlov

$w_1: V \rightarrow R$  modeluje stav uzla (stav batérie, schopnosť spracovávať pakety atď.)

$w_2: E \rightarrow R$  modeluje kvalitu liniek

Cieľ: minimalizácia celkovej použitej energie alebo minimalizácia energie použitej jedným uzlom

Smerovanie: nájdenie najkratšej cesty v  $G$  (vzhľadom na  $w_1, w_2$ ) distribuovaným spôsobom

# Kvalita smerovania

Komunikačná zložitosť: koľko správ je potrebných na poslanie dátového paketa

Dilačný faktor: skutočná komunikačná zložitosť v pomere k najnižšej možnej komunikačnej zložitosti

Cena smerovania: cieľ je najlacnejšia cesta (najkratšia, alebo môže byť použité iné kritérium)

Pamäť: aký priestor potrebujeme na smerovanie

Smerovanie: bez slučiek



# Priemer grafu

Priemer grafu: najdlhšia najkratšia cesta – horný odhad dĺžky spojení v danom grafe

$\text{dist}(u, v)$  = dĺžka najkratšej cesty z  $u$  to  $v$ .

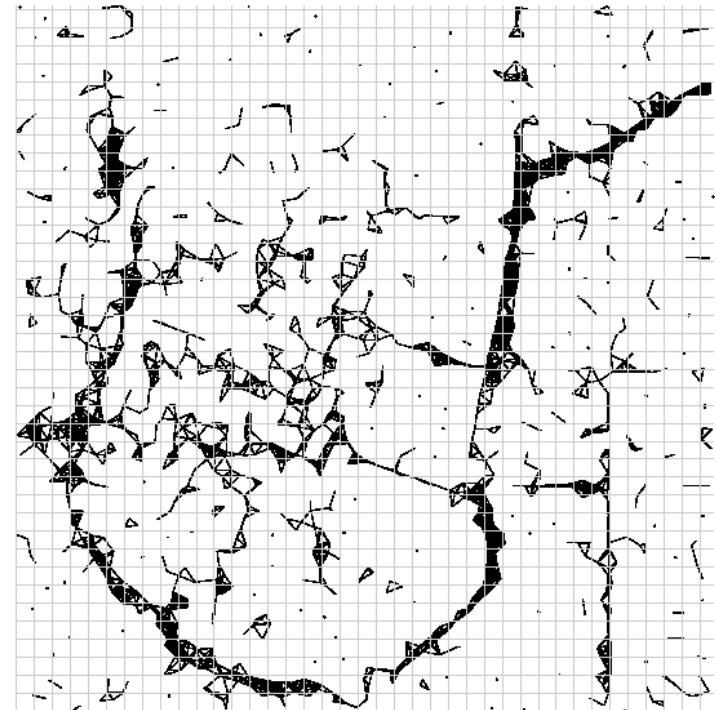
$\text{Diam}(G) = \max \{\text{dist}(u, v)\}$

# Priemer grafu vs d2-párovanie

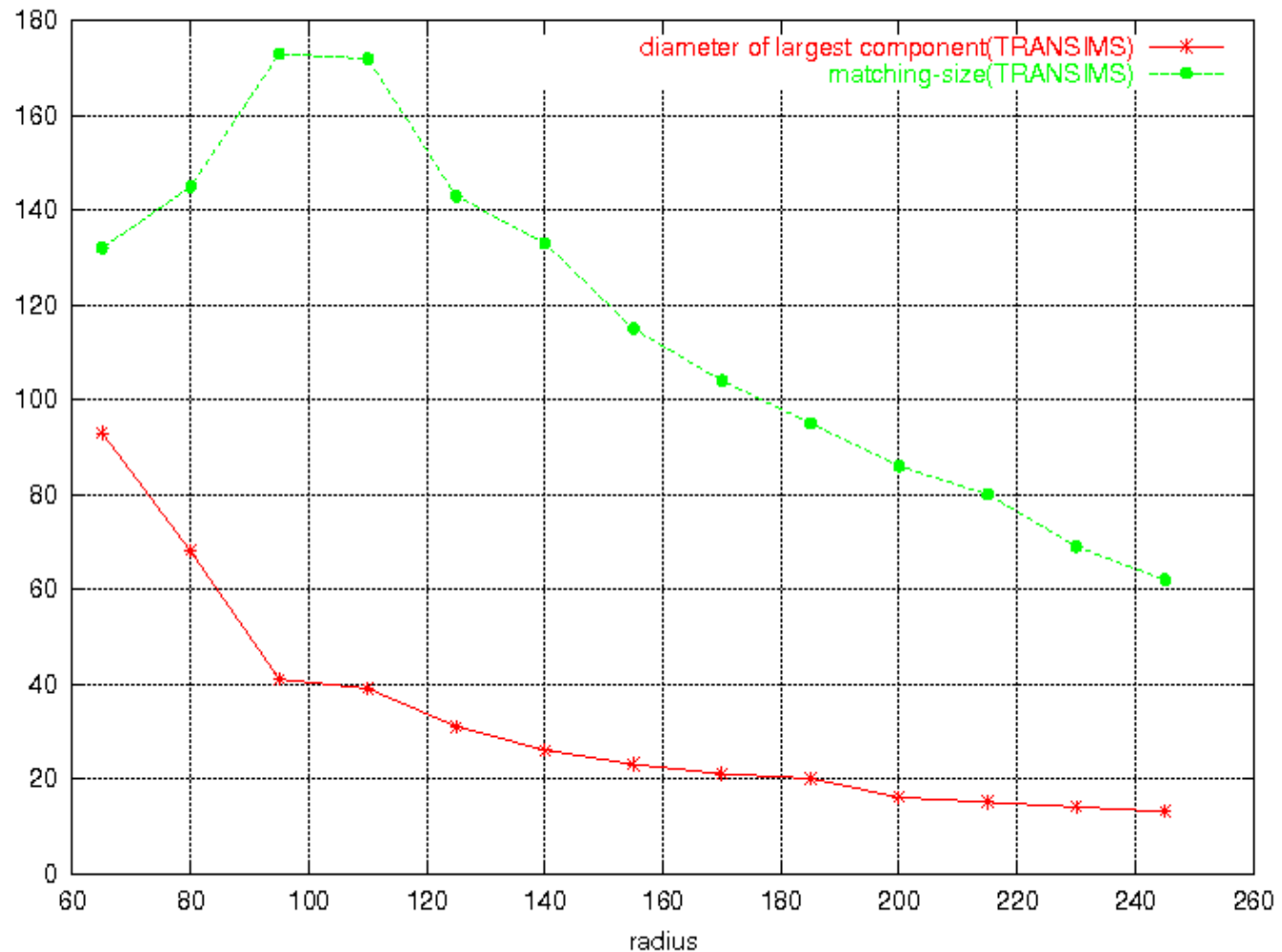
Diam = ?

1718 uzlov

Topológia: Portland, OR



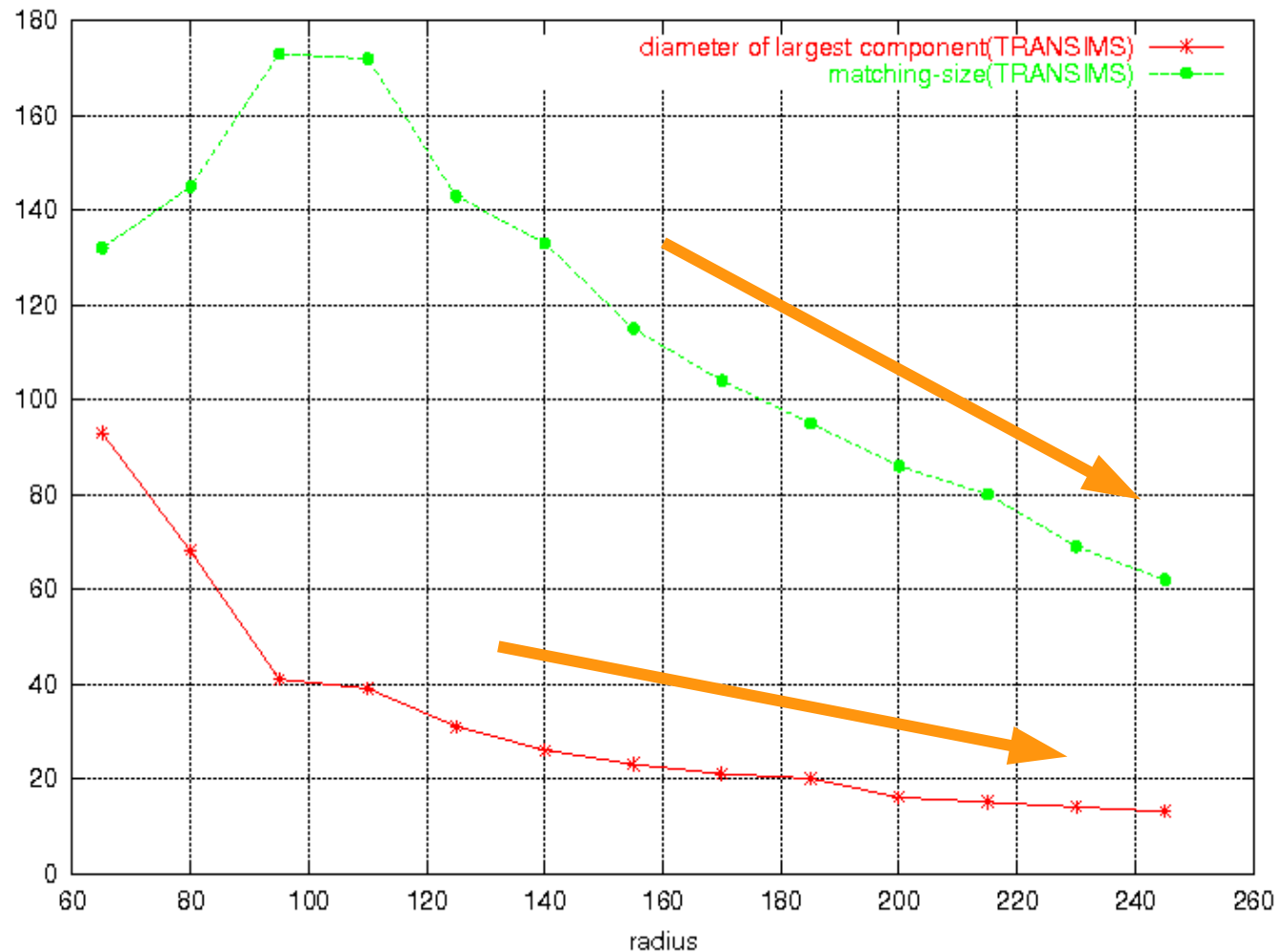
# Priemer grafu vs d2-párovanie



Vieme zvoliť rádius (polomer) tak, aby priemer grafu bol nízky a d2-párovanie vysoké? Ak nie, ktorý parameter je dôležitejší?

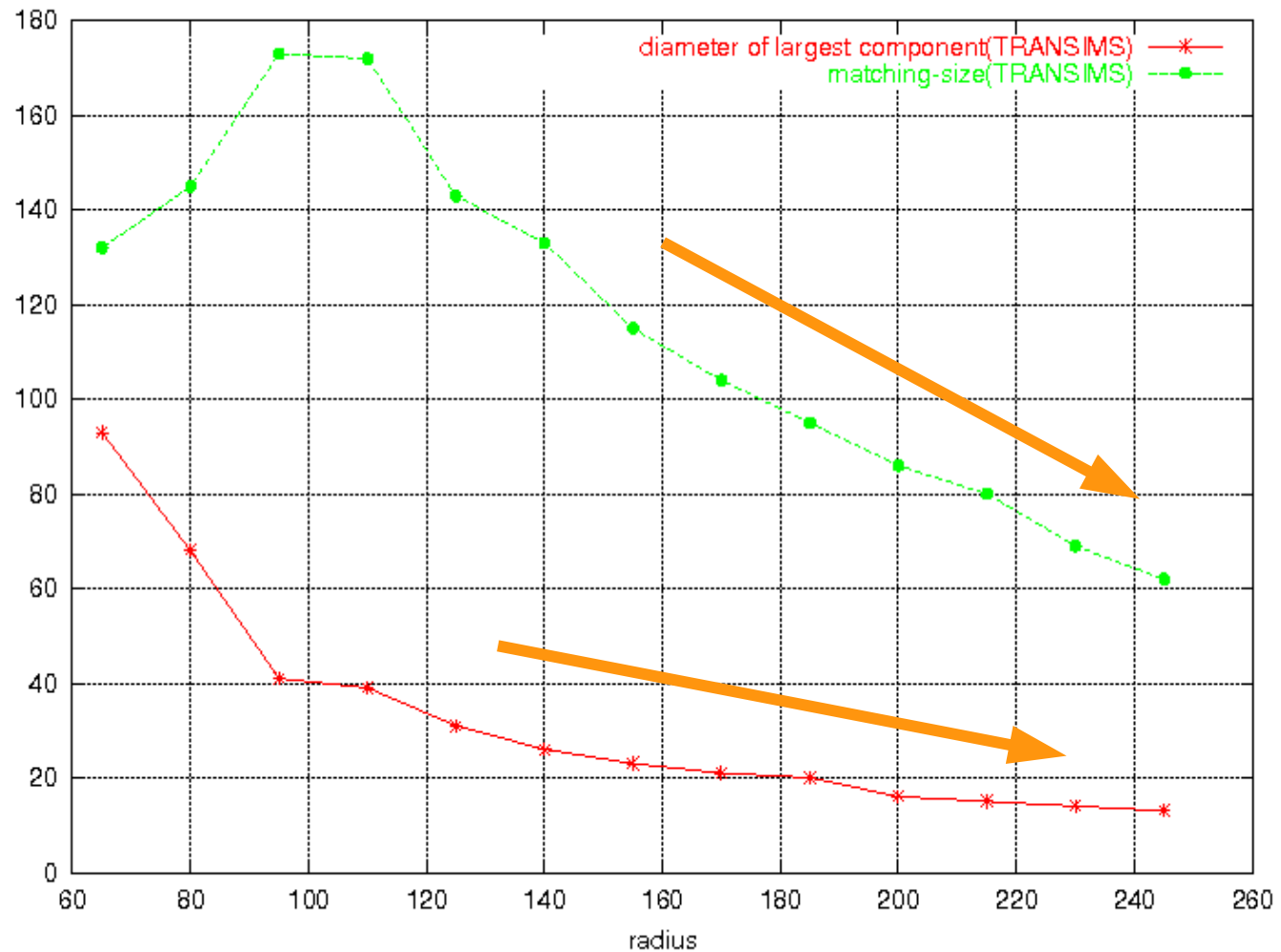
Ak je rádius rovnaký pre všetky uzly, potom možnosti riadenia topológie sú obmedzené.

# Priemer grafu vs d2-párovanie



So stúpajúcim rádiusom klesá priemer, ale zároveň klesá aj hodnota d2-párovania. Z pohľadu d2-párovania je optimálna hodnota rádiusu 90-110m. Z pohľadu priemeru je vhodné zvoliť najväčšiu možnú hodnotu rádiusu. Zariadenia, ktoré umožňujú plynule meniť silu signálu vysielачa sú menej bežné.

# Priemer grafu vs d2-párovanie



d2 párovanie aj priemer je možné vypočítať počas simulácie. V reálnych sieťach to nemusí byť možné (alebo praktické) z dôvodu veľkej dynamiky pohybu uzlov.

# Druhy smerovacích schém

**unicast:** z jedného uzla k jednému uzlu

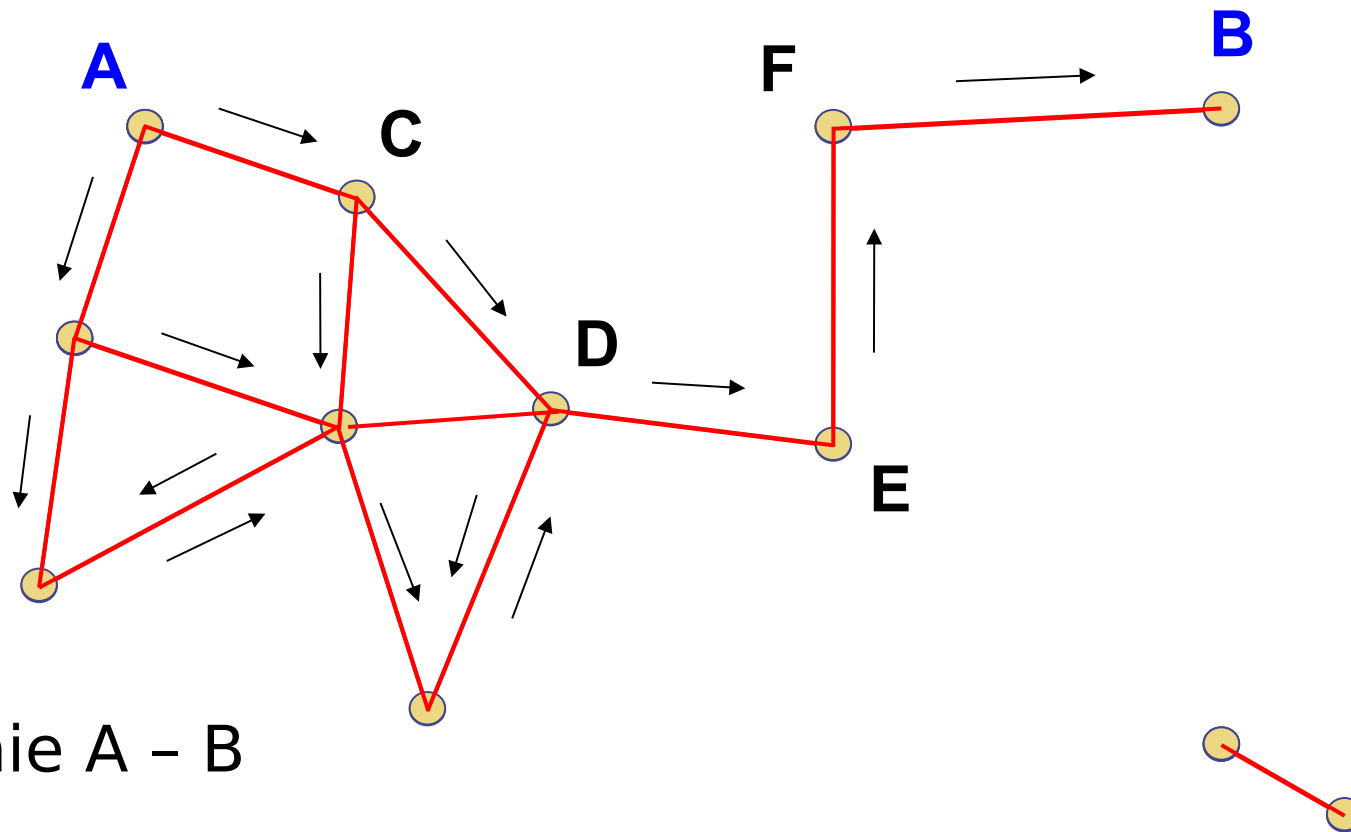
**broadcast:** z jedného uzla ku všetkým uzlom

**convergecast:** z veľa uzlov k jednému uzlu, opak broadcastu, často je posielaný identický paket napr. "echo", alebo je posielaná informácia postupne spracovávaná a konečný uzol už prijme informáciu v agregovanej forme (data fusion)

**multicast:** z jedného uzla k veľa uzlom, alebo naopak.

**anycast:** z jedného uzla k jednému uzlu, ktorý je súčasťou skupiny uzlov (smerovanie k najbližšiemu zo skupiny, ktorý ponúka tú istú službu ako ostatné uzly)

**geocast:** z jedného uzla iným uzlom podľa geografickej polohy



Spojenie A - B

Je možné, že broadcast je použitý ako pomocný mechanismus

**Link state:** každý uzol udrzuje kompletnú vedomosť o sieti tak, že cez broadcast periodicky posiela informáciu o svojich susedoch, kvalite pripojenia k nim. Uzol, ktorý takúto informáciu dostane aktualizuje svoju smerovaciu tabuľku a prepočíta vzdialenosti k všetkým ostatným uzlom.

**Distance-vector routing:** každý uzol udrzuje vedomosť o vzdialenostiach k všetkým ostatným uzlom a o susednom uzle, ktorý leží na najkratšej ceste k cieľovému uzlu. Úplná cesta k cieľovému uzlu nie je známa (okrem prípadu, keď je cieľový uzol zároveň aj susedný uzol).



# Smerovanie: link state a distance-vector

**Link state:** v hlavičke paketa je kompletná cesta do cieľového uzla.

**Distance-vector routing:** v hlavičke paketa je id nasledujúceho uzla, tento nasledujúci uzol rozhodne kam bude paket v ďalšom kroku preposlaný.

# Smerovanie: proaktívne a reaktívne

**Proaktívne protokoly** udržujú cesty k všetkým ostatným uzlom bez ohľadu na to, či nejaká cesta bude v budúcnosti potrebná.

- DSDV

**Reaktívne protokoly** vypočítajú cestu k cieľovému uzla až keď je potrebná.

- AODV, DSR

**Hybridné protokoly**

# Smerovanie: reaktívne

**Zdrojové smerovanie:** kompletná cesta do cieľového uzla je v hlavičke paketa. Nevýhoda je narastanie veľkosti paketa s dĺžkou cesty.

**Skok po skoku (hop-by-hop):** v hlavičke paketa je len id nasledujúceho uzla a id cieľového uzla. Nevýhoda je, že každý uzol musí mať smerovaciu informáciu, v prípade, že takáto informácia neexistuje, je potrebný výpočet smerovania z tohto uzla do cieľového uzla.

# DSDV (Destination-sequenced distance vector)

Proaktívne hľadanie ciest

Distance-vector smerovanie

Aktualizácia smerovacej informácie:

- full dump: kompletná smerovacia informácia
- Inkrementálna: len zmeny od poslednejho “full dump”, častejšie ako “full dump”.

Overhead (aktualizácia):  $O(n^2)$

Memory overhead:  $O(n)$

Založené na Bellman-Fordovom algoritme.

# DSDV (Destination-sequenced distance vector)

Aktualizácie:

- **Obsahujú sekvenčné číslo (logický čas)**, ktorý umožňuje rozlišovať medzi aktualizáciami
- id cieľového uzla
- Vzdialenosť cieľového uzla

Reaktívny protokol.  
Distance-vector smerovanie.  
Hop-by-hop smerovanie.

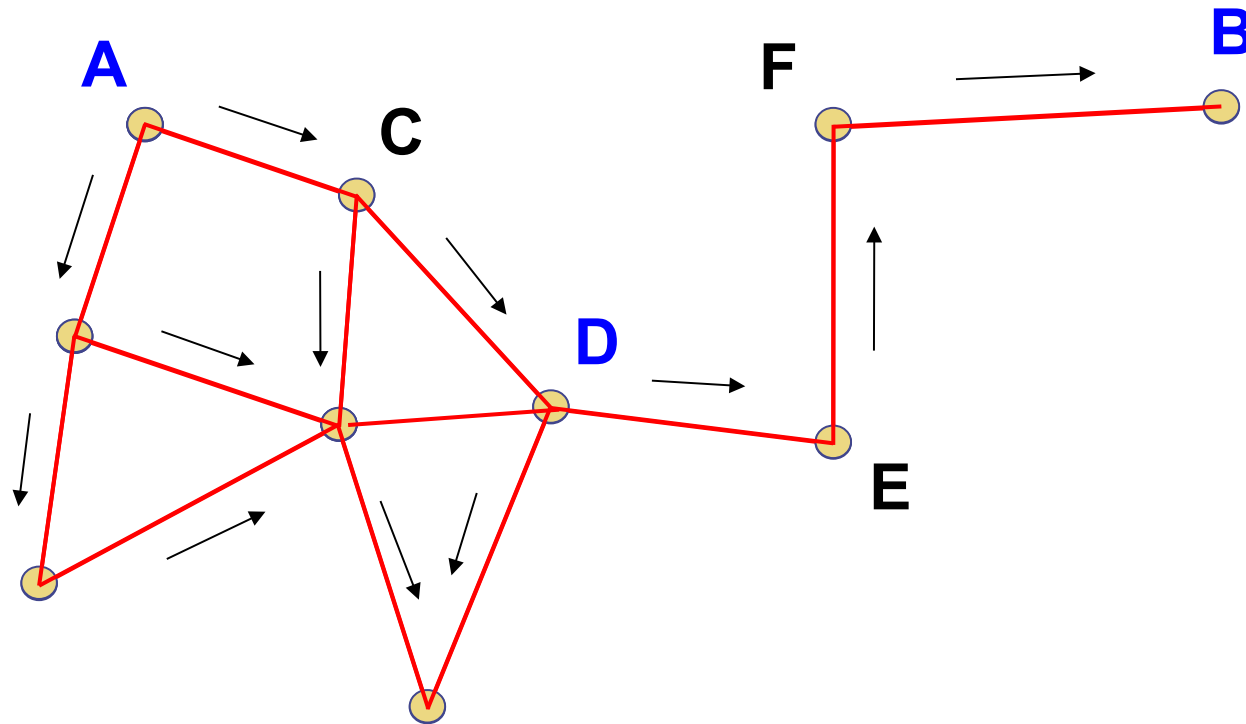
Využíva:

- RREQ: route request
- RREP: route reply
- RERR: route error

Hľadanie cesty:

- Uzol potrebuje poslať dátový paket a cesta k cieľovému uzlu nie je známa, potom cez broadcast pošle RREQ, každý RREQ má sekvenčné číslo.
- Len prvý prijatý RREQ s daným sekvenčným číslom je preposlaný cez broadcast.
- (odpoveď) Uzol, ktorý pozná cestu k cieľovému uzlu, alebo samotný cieľový uzol, odpovedajú s RREP na prijatý RREQ. RREP je poslaný cez unicast.

## RREQ-RREP: príklad



*A potrebuje cestu k B.*

RREQ je poslaný cez broadcast.

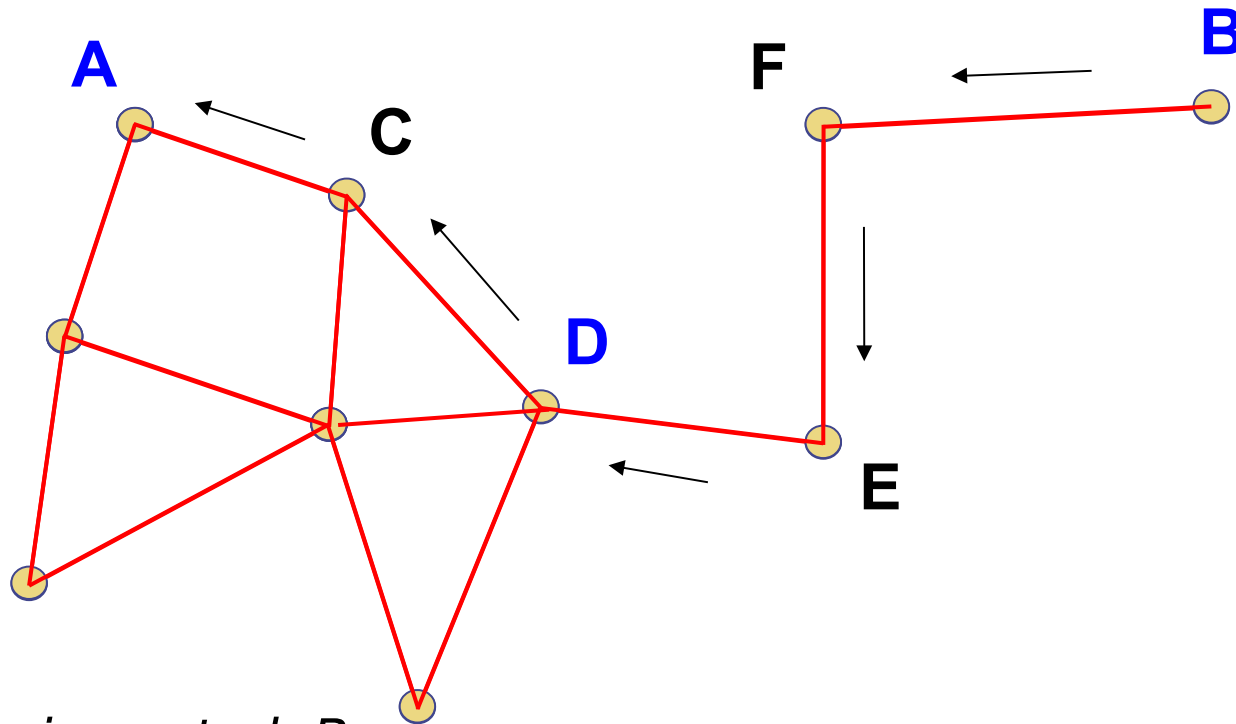
# RREQ-RREP: príklad

## *Odpoved':*

- Uzol, ktorý pozná cestu k cieľovému uzlu, alebo samotný cieľový uzol, odpovedajú s RREP na prijatý RREQ. RREP je poslaný cez unicast.
- Zdrojový uzol môže dostať niekoľko RREP a môže sa rozhodnúť, ktorou cestou budú posielané pakety (najkratšia cesta je default).



## RREQ-RREP: príklad



*A potrebuje cestu k B.*

1: D pozná cestu k B a odpovie s RREP.

2: RREQ je preposlané až k cieľovému uzlu B, ktorý odpovie s RREP.

# AODV: expanding ring

Expanding ring je optimalizácia, ktorá umožňuje preposlanie RREQ len do určitej vzdialenosti od zdrojového uzla.

- zdrojový uzol nastaví  $TTL = TTL\_START$  a nastaví time-out pre prijatie RREP na  $RING\_TRAVERSAL\_TIME = 2 * TTL * NODE\_TRAVERSAL\_TIME$  (ms)
- ak zdrojový uzol do time-outu neprijme RREP, potom zvýši TTL o  $TTL\_INCREMENT$ . Takéto zvyšovanie pokračuje až po dosiahnutie  $TTL\_THRESHOLD$ .
- Nasleduje nastavenie  $TTL = NET\_DIAMETER$ , posielanie RREQ je opakované  $RREQ\_RETRIES$  krát.

Prednastavené hodnoty:

TTL_START	1
TTL_INCREMENT	2
TTL_THRESHOLD	7
NODE_TRAVERSAL_TIME	40 [ms]
NET_DIAMETER	35
RREQ_RETRIES	2

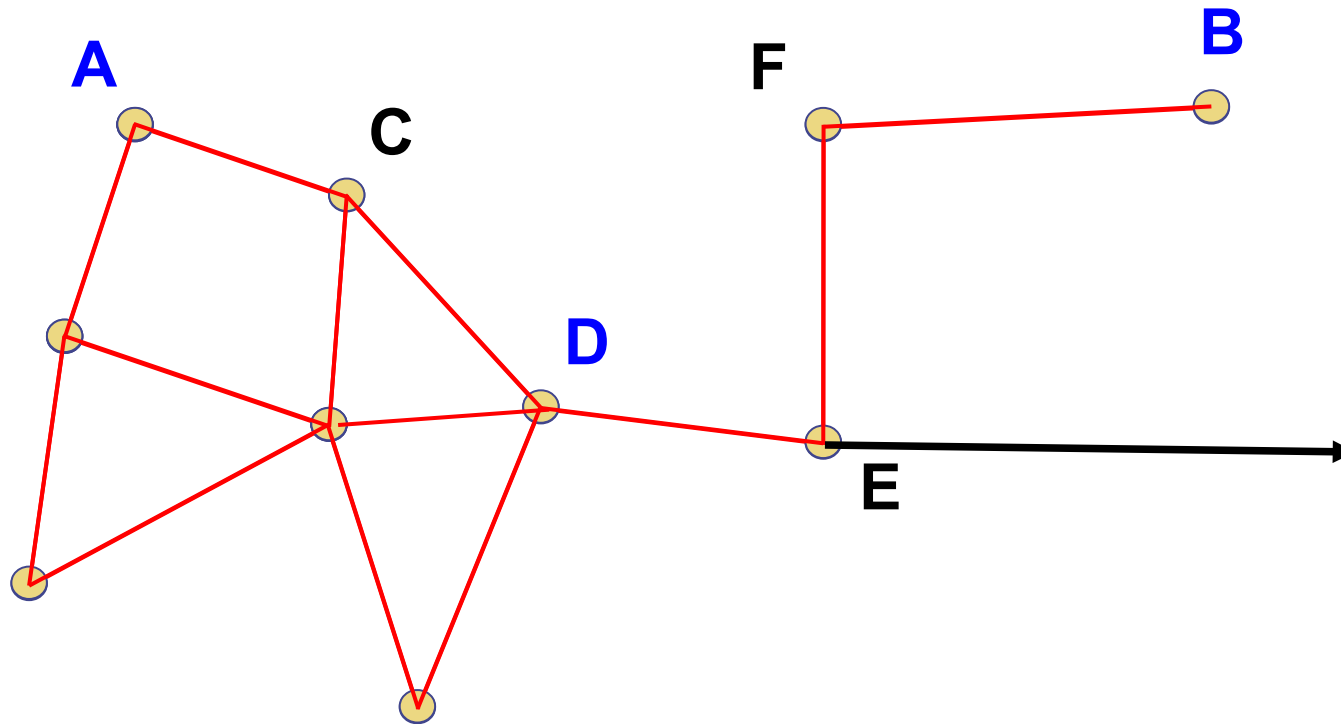
TTL = time-to-live, informácia je vpísaná do hlavičky RREQ

RERR je poslaný uzlom, ak jeho sused neodpovedá.

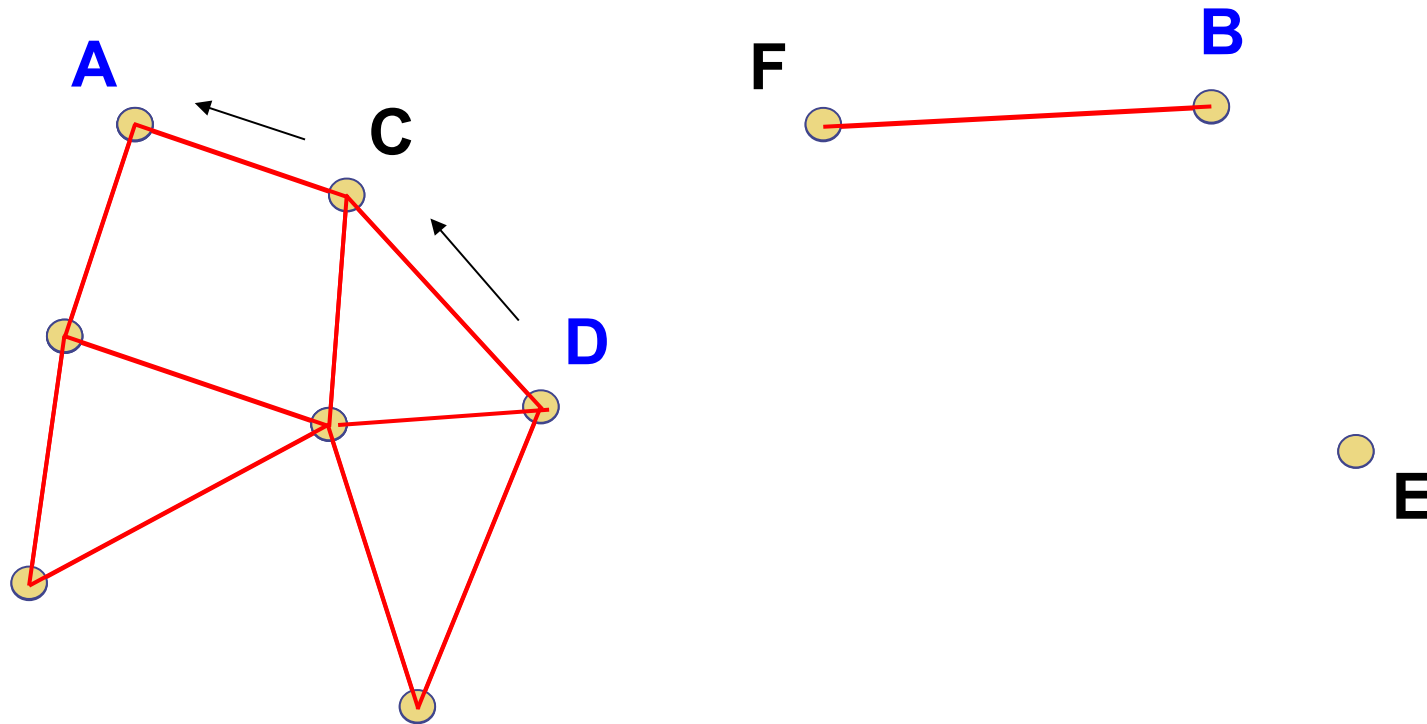
RERR je druh údržby cesty.

Implementácia:

- Prepojenie s MAC protokolom, keď na RTS nie je odpovedané s CTS/DATA/ACK.
- Pomocou periodických HELLO paketov (overhead).



*E sa pohne preč.*



D pošle RERR paket. A sa pokúsi nájsť novú cestu k B. Ak to nie je možné, potom už nebude ďalej posielat' dátové pakety (uzlu B).

# AODV: expirácia ciest

- Nájdená cesta ostane platná pokiaľ je využívaná.
- Ak cesta nie je využívaná potom je zmazaná (prednastavená hodnota `ACTIVE_ROUTE_TIMEOUT` je 3000ms).

# DSR (Dynamic source routing)

Reaktívny protokol.

Zdrojové smerovanie t.j. kompletná cesta v hlavičke paketu.

Využíva RREQ, RREP, RERR identickým spôsobom ako AODV.

# DSR: optimalizácia

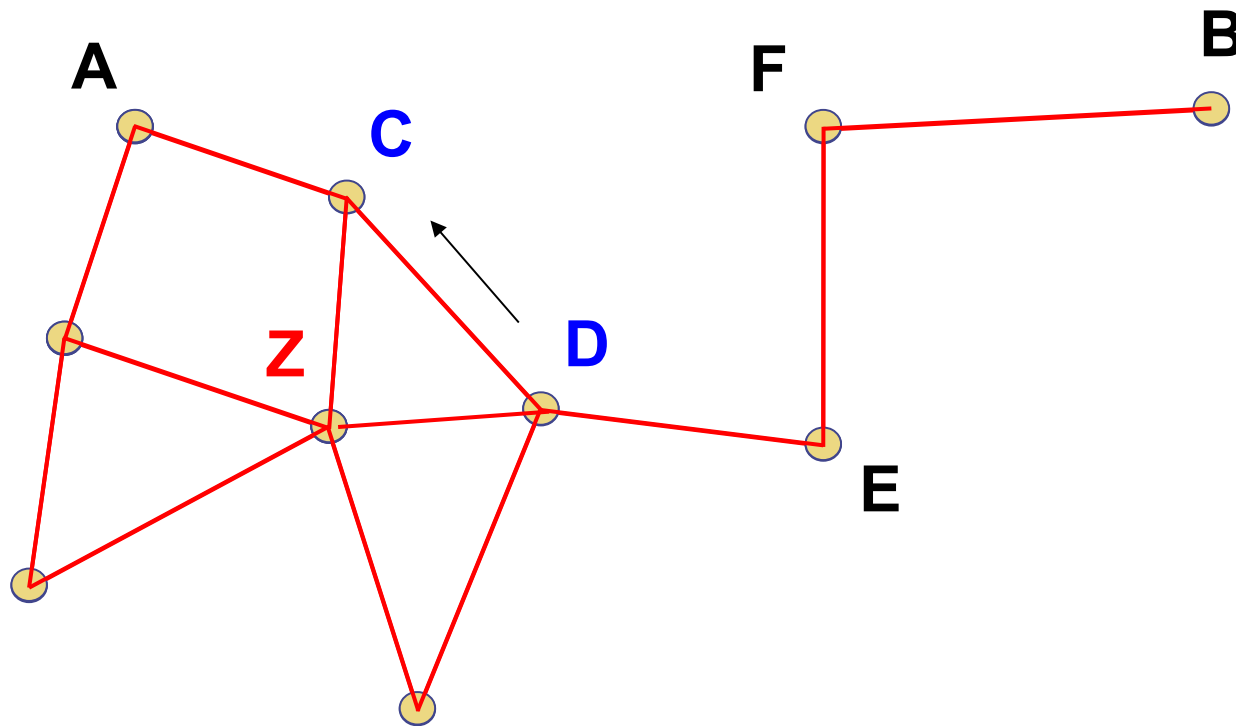
Packet salvaging: ak je preposlanie paketu nemožné, uzol hľadá alternatívnu cestu s RREQ-RREP. Zdrojový uzol bude o zmene informovaný pomocou RERR.

RERR piggybacking: RERR sú pripojené k RREQ. Týmto sa informácia o chybe rýchlejšie šíri.

Automatic route shortening: ak uzol zistí, že existuje skrátenie cesty (toto je možné pri zdrojovom smerovaní).



# Promiscuous mode



Uzol Z prepočuje výmenu informací mezi D a C.  
Promiscuous mode je energeticky náročný.

Komunikačná zložitosť:

Získanie cesty:  $O(n)$

Údržba cesty:  $O(n)$

$n$  je počet uzlov.

DSR + GPS information.

(x,y,z)-koordináty získané cez GPS

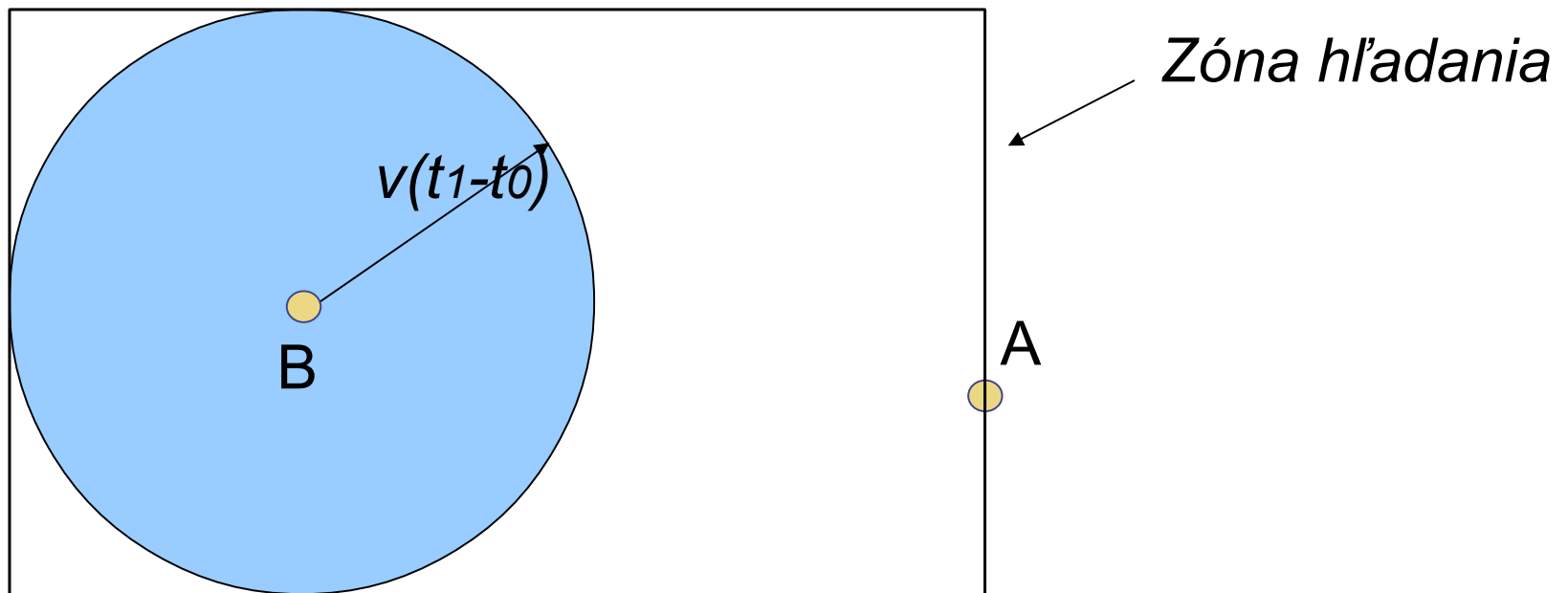
**Očakávaná zóna:** odhad kde by sa uzol mal nachádzať.

**Zóna hľadania:** najmenší obdĺžnik, ktorý obsahuje očakávanú zónu. Len v zóne hľadania je možné preposlať RREQ.

# LAR scheme 1 (Location aided routing)

Uzol B bol v čase  $t_0$  na pozícii  $(x,y,z)$  a pohyboval sa rýchlosťou  $v$ . Teraz je čas  $t_1$ .

Očakávaná zóna: kruh s centrom  $(x,y,z)$  a polomerom  $v(t_1-t_0)$ .



# LAR scheme 1 (Location aided routing)

RREQ sú preposielané len v zóne hľadania.

RREP obsahuje GPS informáciu cieľového uzla.

Výhoda: škálovateľnosť s počtom uzlov.

Nevýhoda: GPS je energeticky náročné, a teda LAR schme 1 nie je vhodné smerovanie pre zariadenia napájané z batérie.

RREQ, RREP, RERR z pohľadu MAC vrstvy sú dátové pakety.

Dôsledok: detekcia nosnej, RTS-CTS-DATA-ACK sú použité.

Broadcast používa len detekciu nosnej. Prijatie CTS od všetkých susedov nie je možné. Nie je možné poslať paket veľkému množstvu susedov s potvrdením prijatia.

Amit Dua, Neeraj Kumar, Seema Bawa, A systematic review on routing protocols for Vehicular Ad Hoc Networks, Vehicular Communications, Volume 1, Issue 1, January 2014, Pages 33-52, ISSN 2214-2096, <http://dx.doi.org/10.1016/j.vehcom.2014.01.001>.

# Live Q&A

Live otázky a odpovede sú plánované cez Discord.