

# Maršrutizācija bezvadu sensoru tīklos

Ģirts Straziņš

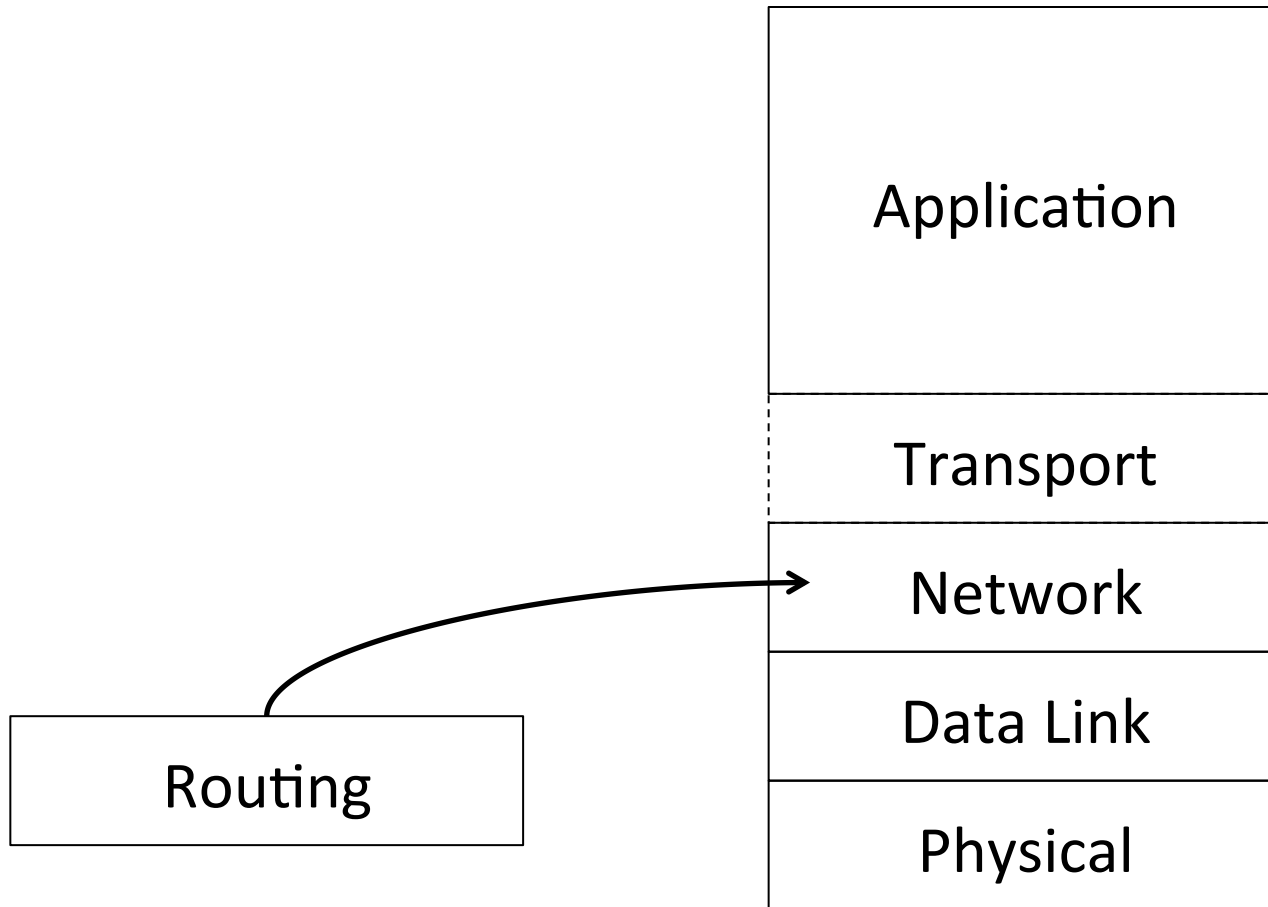
Kurss “Bezvadu sensoru tīkli” [B]

Datorikas fakultāte

Latvijas Universitāte

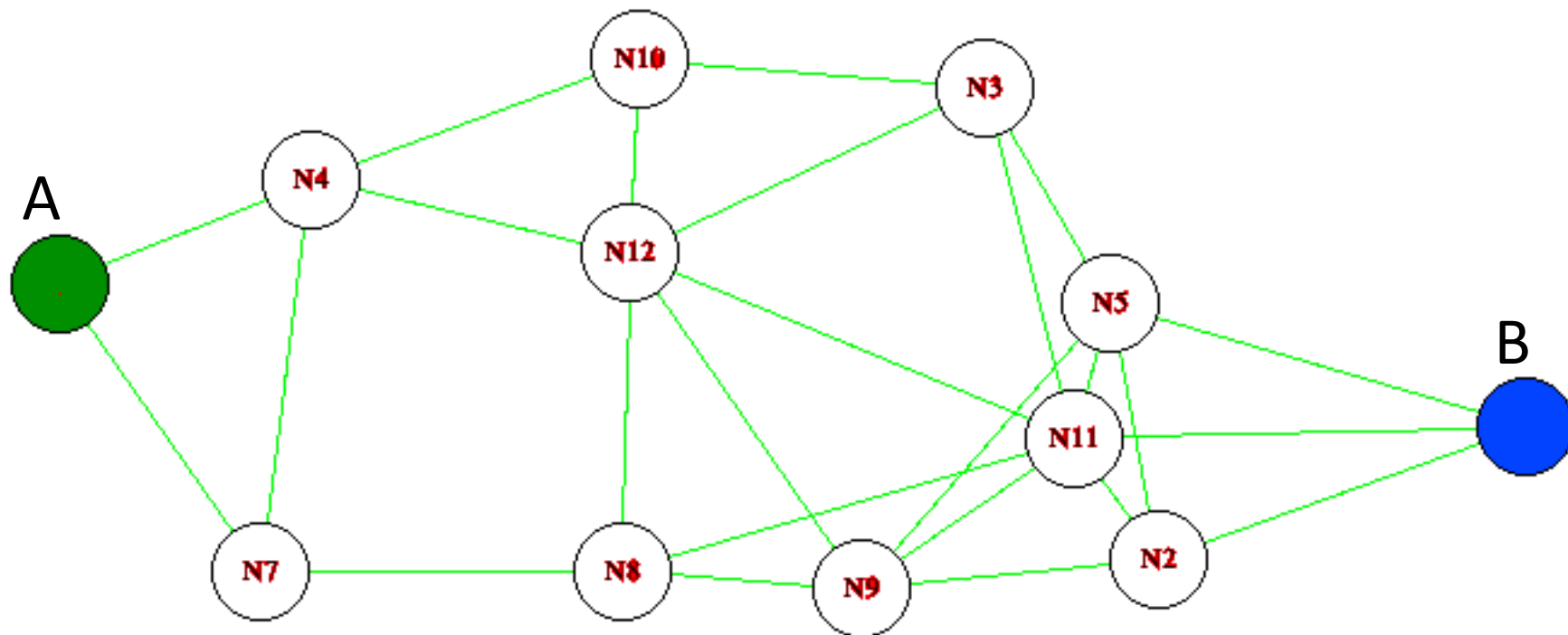
19.okt.2011.

# Atkal par OSI līmeņiem



# Maršrutizācijas uzdevums

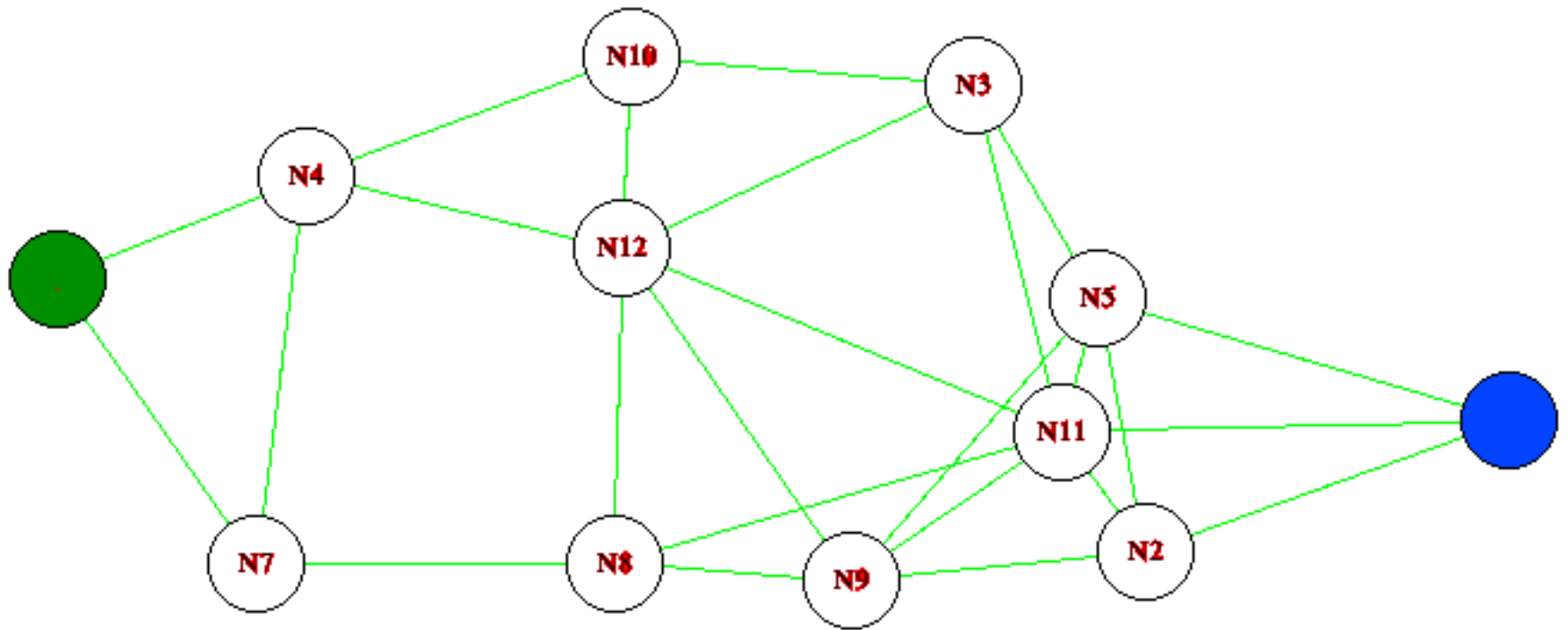
- Nogādāt datus no A uz B



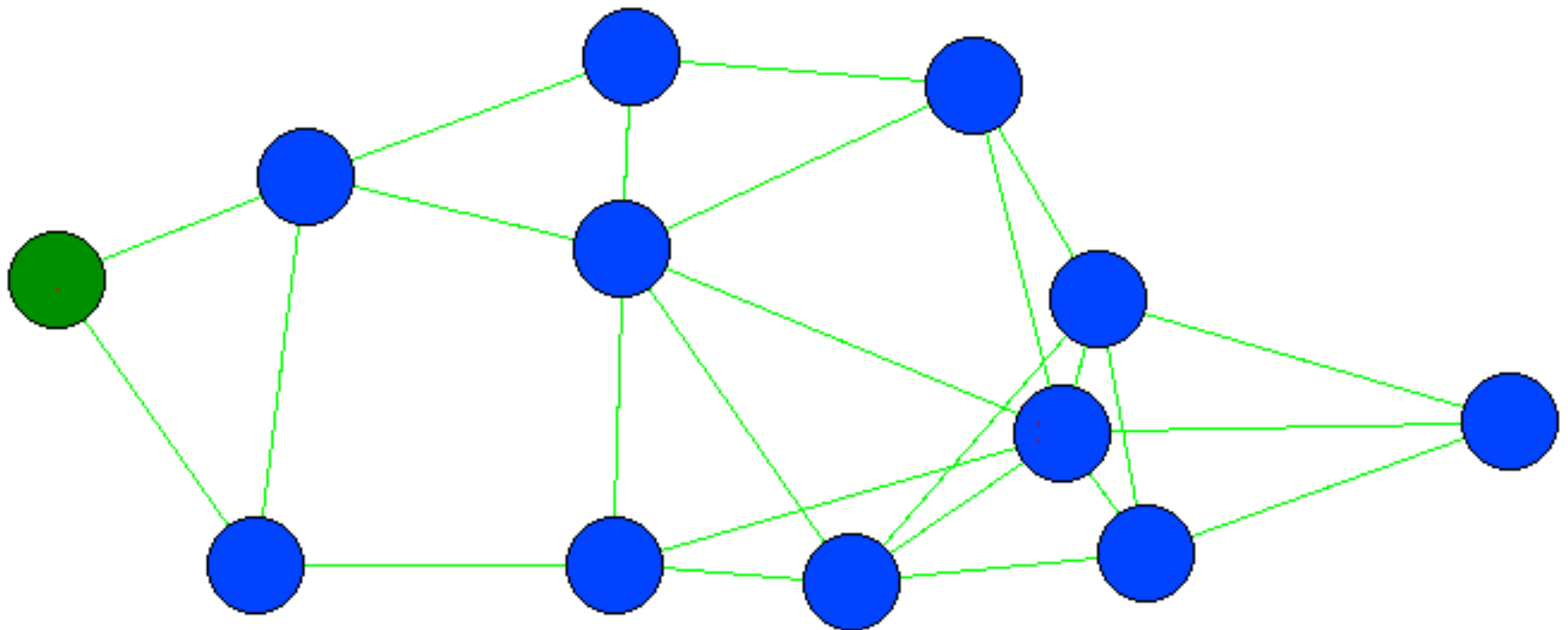
MAC ≈ Maršrutizācija



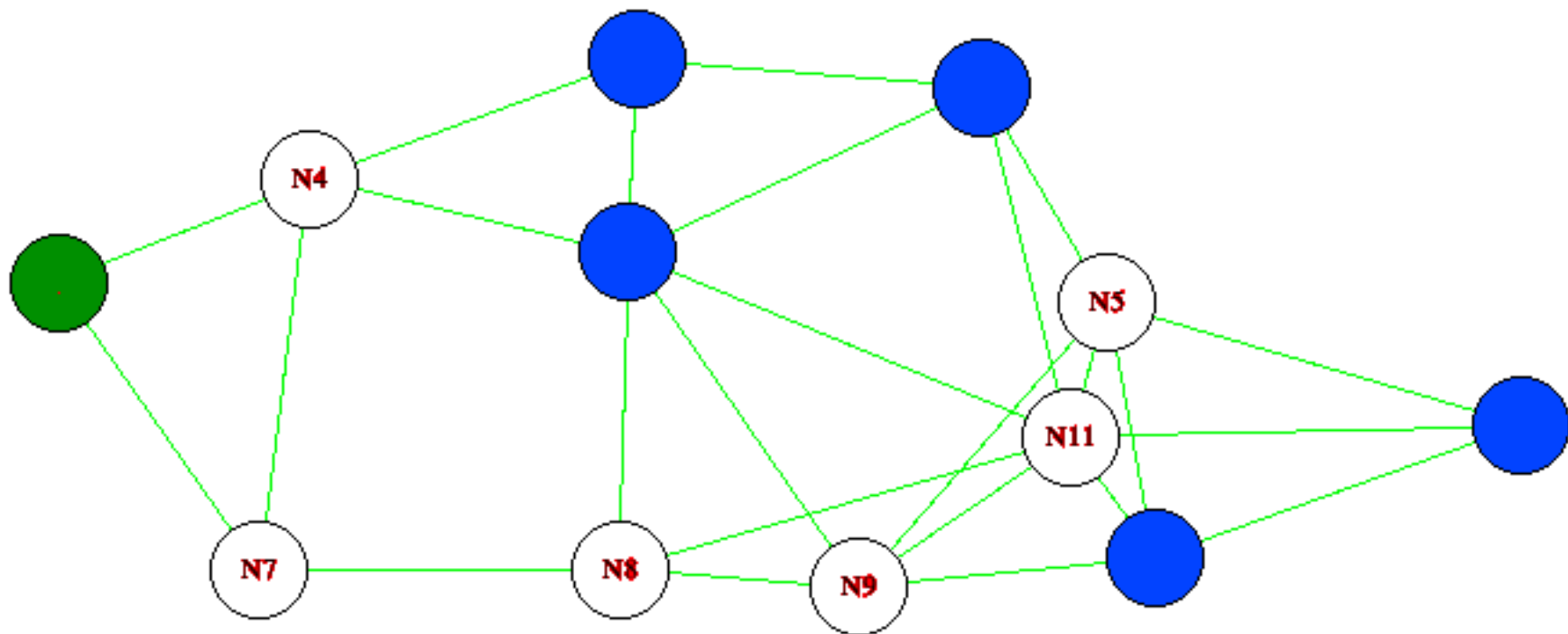
# Saņēmēju klase Unicast



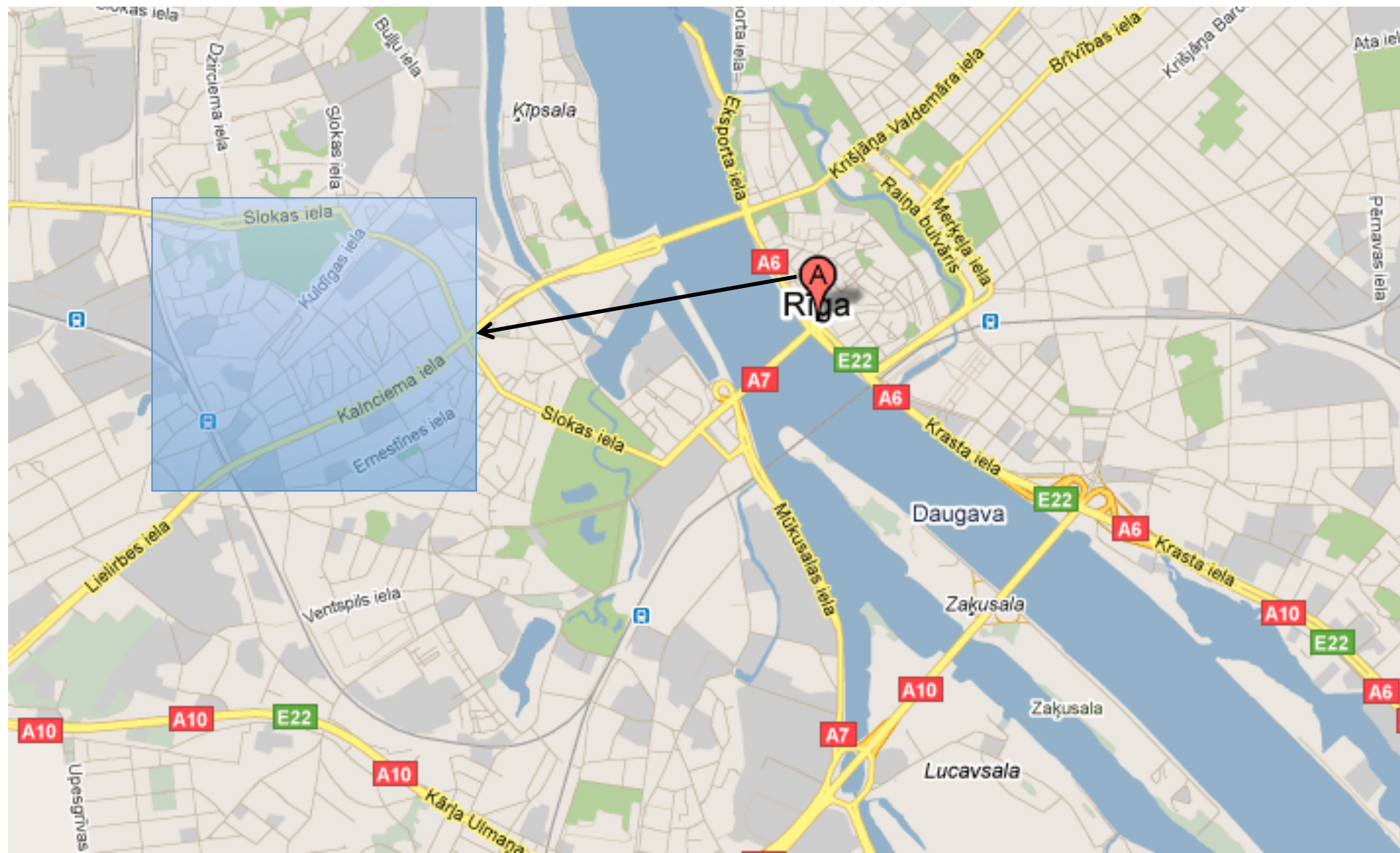
# Saņēmēju klase Broadcast



# Saņēmēju klase Multicast



# Specifiska klasse *Geocast*



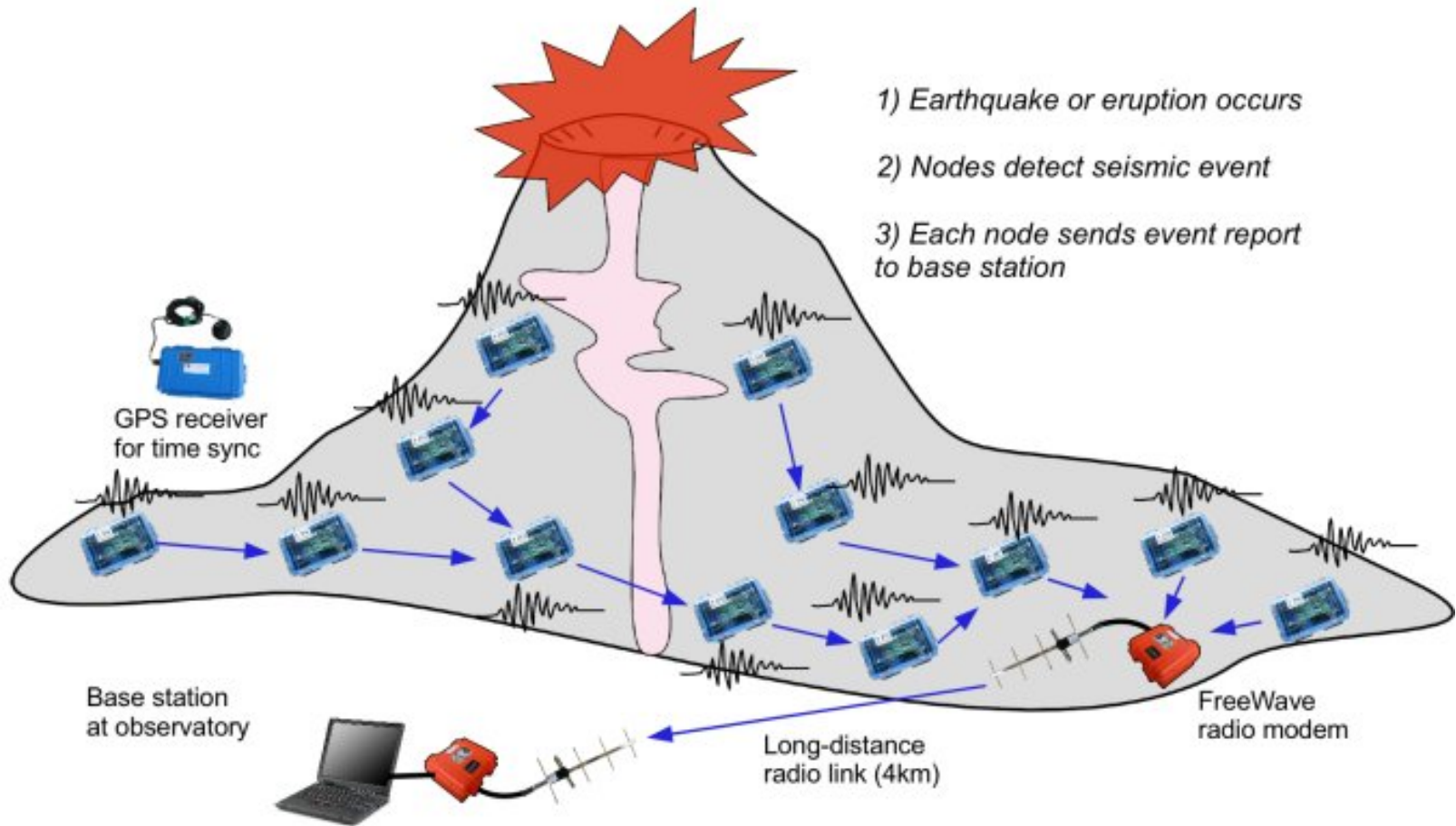


Kādas būtu prasības ideālai  
maršrutizācijai?

# Ideālā maršrutizācija

- Pilns maršrutizācijas grafs
- Mazs servisa informācijas apjoms
- Atšķirīgu tīkla mezglu atbalsts
- Noturība pret negadījumiem
- Mērogojamība
- Dinamiskas topoloģijas atbalsts
- QoS nodrošināšana
- Datu agregāciju atbalstoša topoloģija
- ...

# Kādas papildus prasības ir maršrutizācijai **sensoru tīklos**?

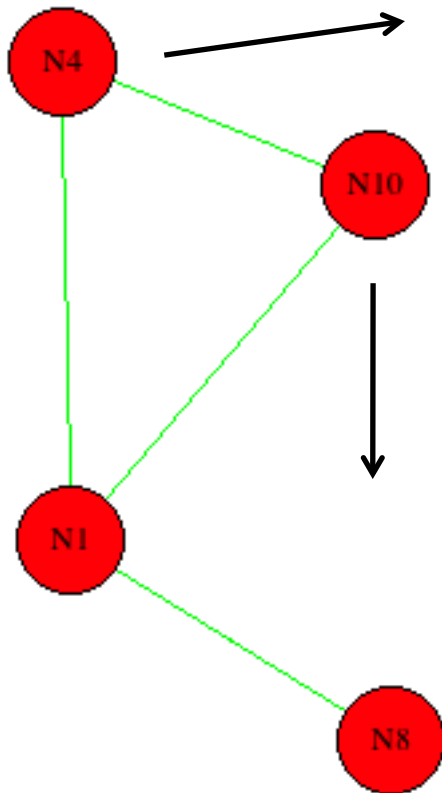


# BST maršrutizācijas specifika

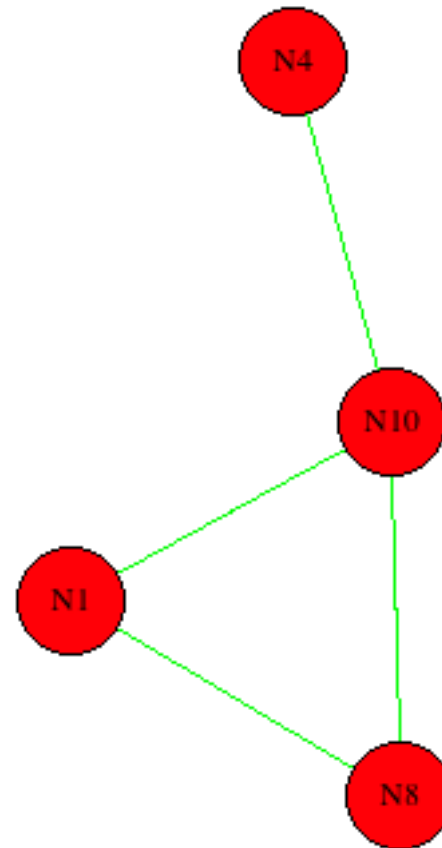
- Energo efektivitāte
- Mazi komunikācijas attālumi
- Daudz mezglu
- Mazs atmiņas apjoms un veikspēja
- Bezvadu sakari
- Reāla vide, ne servertelpas

# Mobilu tīklu specifika

Pirms

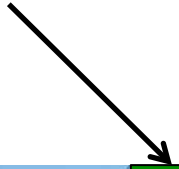


Pēc

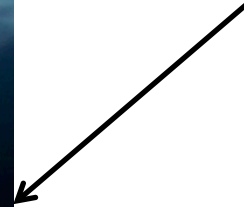


# Tur ārā topoloģija ir mainīga

Tā ir mote



Te ir motes,  
kas nevar  
sarunāties

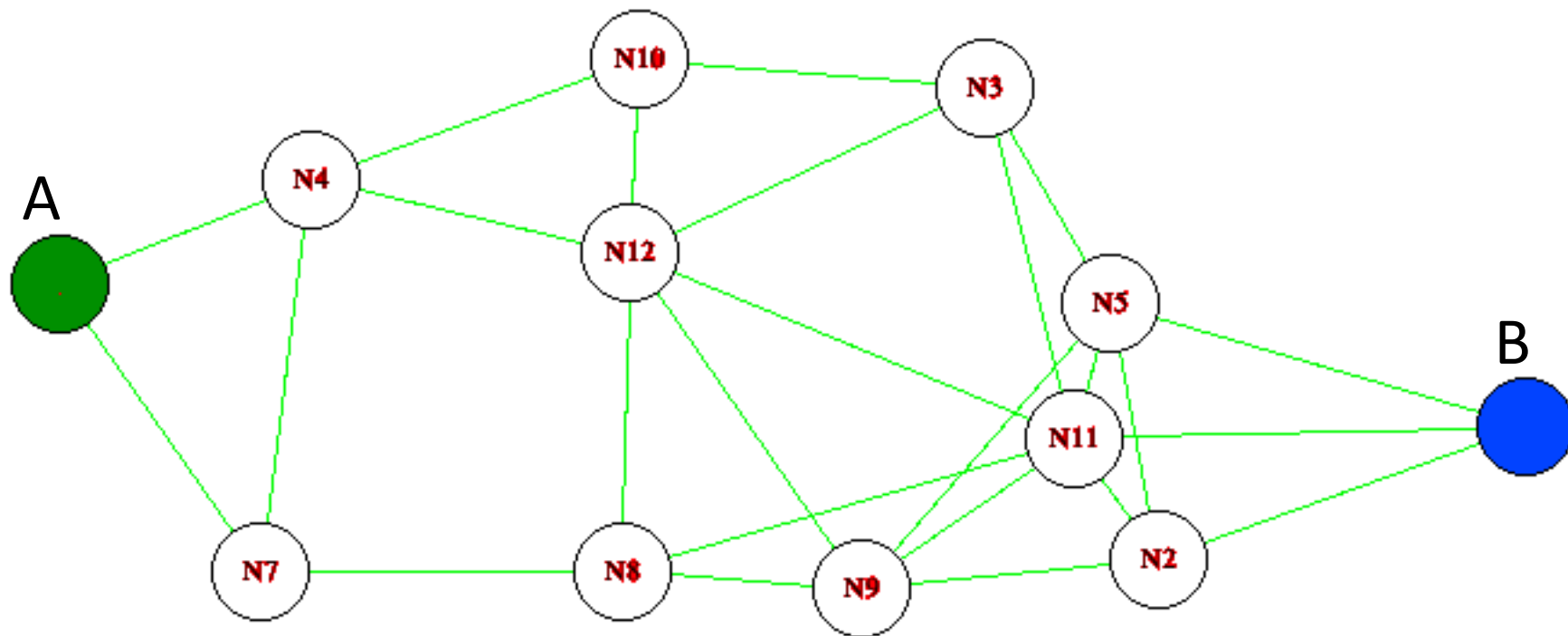


Tur bija mote



# Kā glabāt maršrutizācijas info?

- Kā A mezglam atcerēties ceļu uz B?



# Parasti pietiek ar kaimiņu

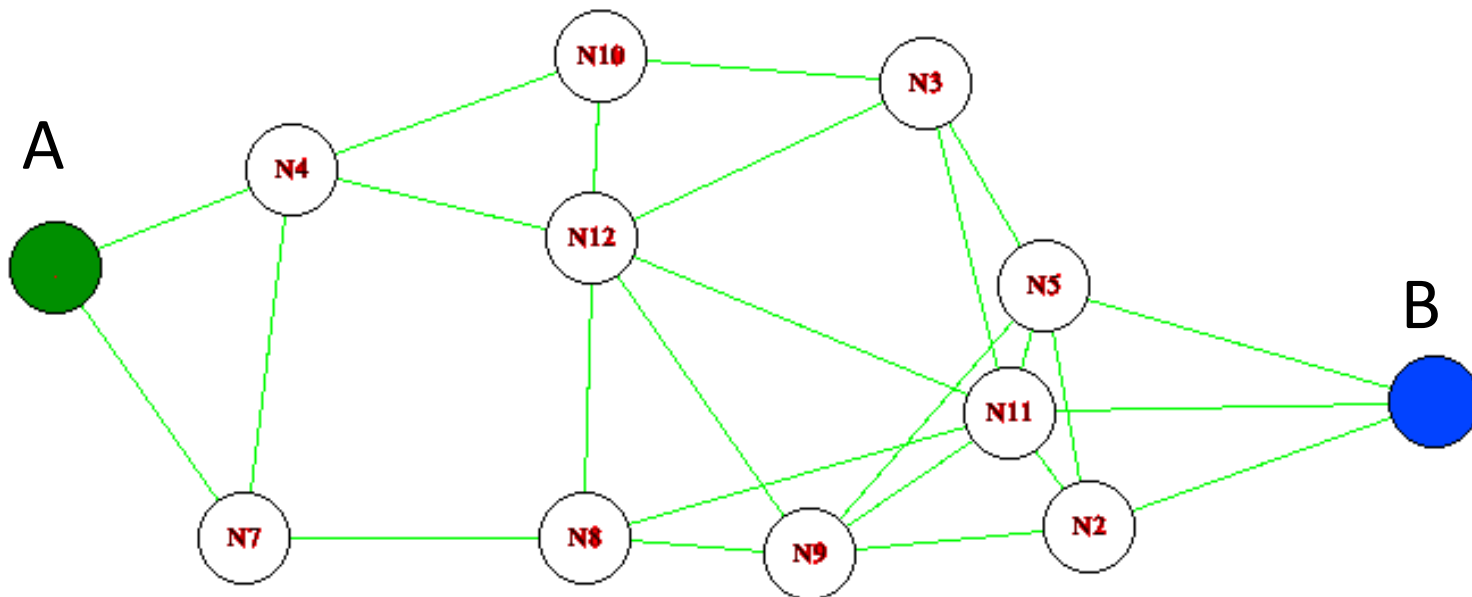
- Atcerēties, caur kuru kaimiņu (nexthop) jāsūta uz B





# Maršrutizācijas tabula

Destination	Nexthop	Hop Count
N2	N7	4
N4	N4	1
N8	N7	2
B	N4	4



# Ģeogrāfiskā maršrutizācija

- Nav nepieciešama maršrutizācijas tabula
- Bet nepieciešams zināt mezglu pozīcijas
- Strādā labi, ja jāsūta dati uz noteiktu apgabalu (geocast), nevis konkrētam mezglam (unicast)

Ko izmantot par saites metriku?

# Tipiskās savienojuma metrikas

- Uztvertā signāla stiprums
- Sūtīšanas izmaksas (enerģija)
- Fiziskais attālums
- Sūtīšanas ilgums

# Proaktīva pieeja

- Savākt visu jau sākumā – ja nu vajadzēs



# Reaktīva pieeja

- Kad vajadzēs, savākšu



# Kas labāks?

Proaktīvi

Reaktīvi

6'0"

5'6"

5'0"

4'6"

4'0"

3'6"

3'0"



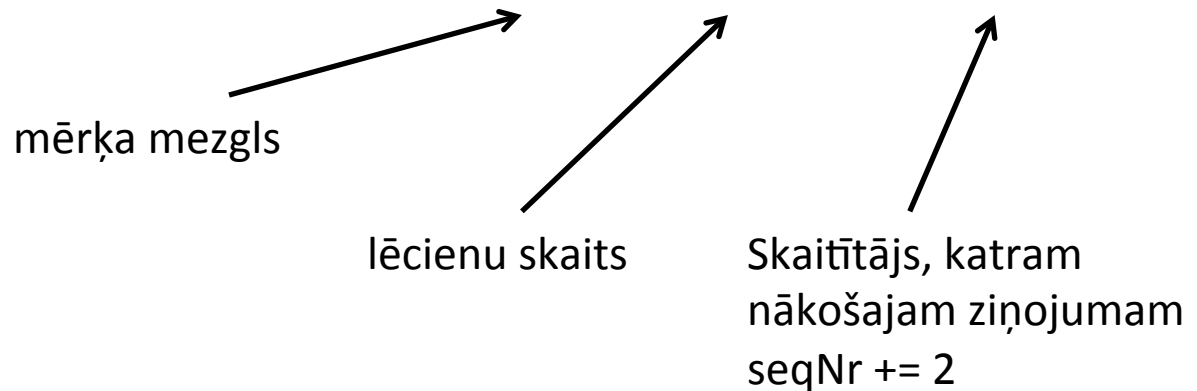
# The Usual Suspects



# DSDV, 1994.g

Destination Sequenced Distance Vector

- Mezgli periodiski sūta (dest, hops, seqNr)



- seqNr vienmēr pāra skaitlis
- Izmanto maršrutu ar lielāko seqNr
- Ja savienojums pazūd, (dest,  $\infty$ , seqNr+1)

# DSDV animācija

<http://www.dpunkt.de/mobile/code/dsdv.html>

# AODV, 2003.g

Ad-hoc On-demand Distance Vector

- Balstās uz DSDV, bet reaktīvs
- Izsūte RouteRequest, saņem RouteReply
- Ja maršruts pazūd, saņem RouteReply

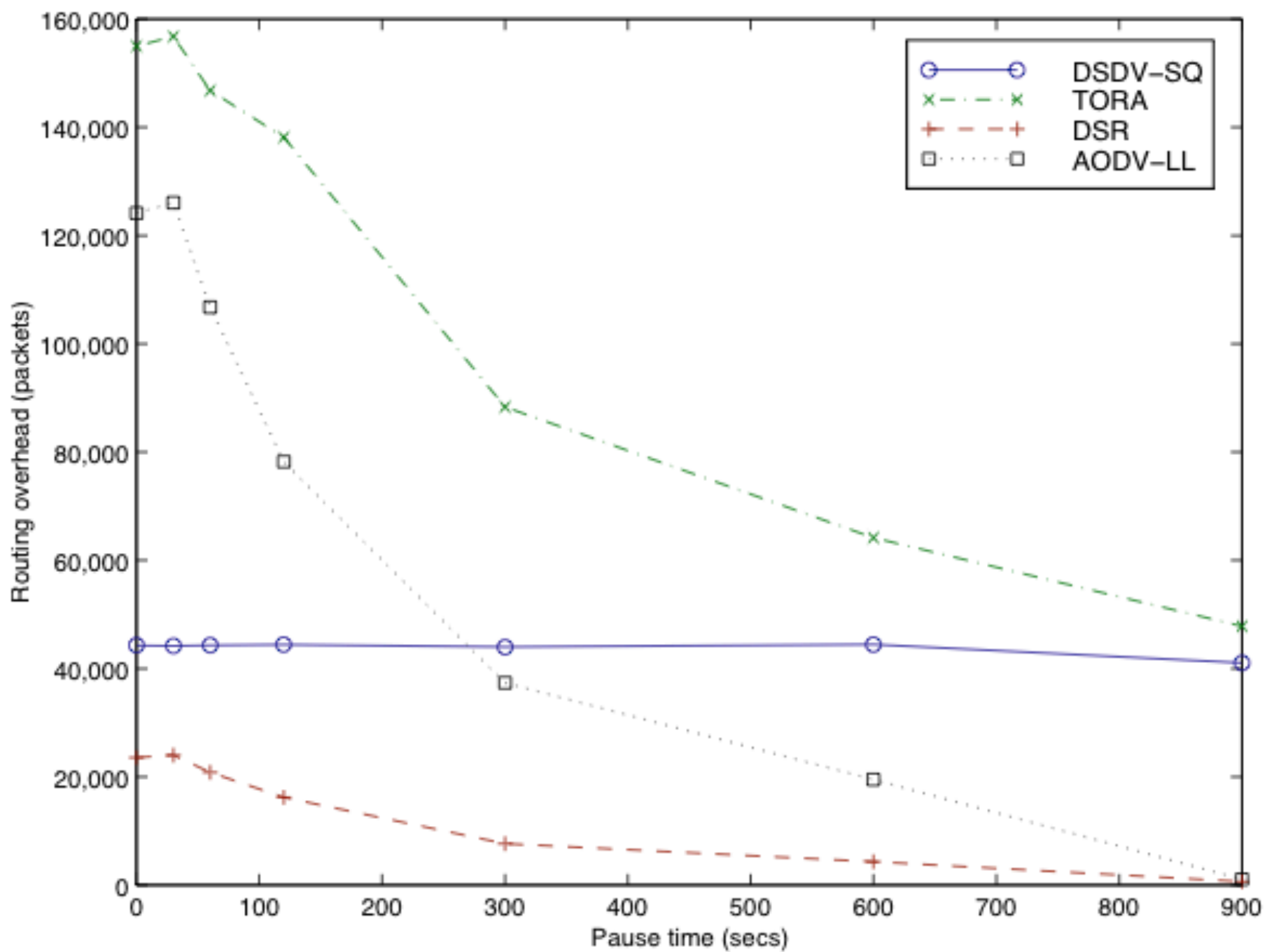
# DSR, 1996.g

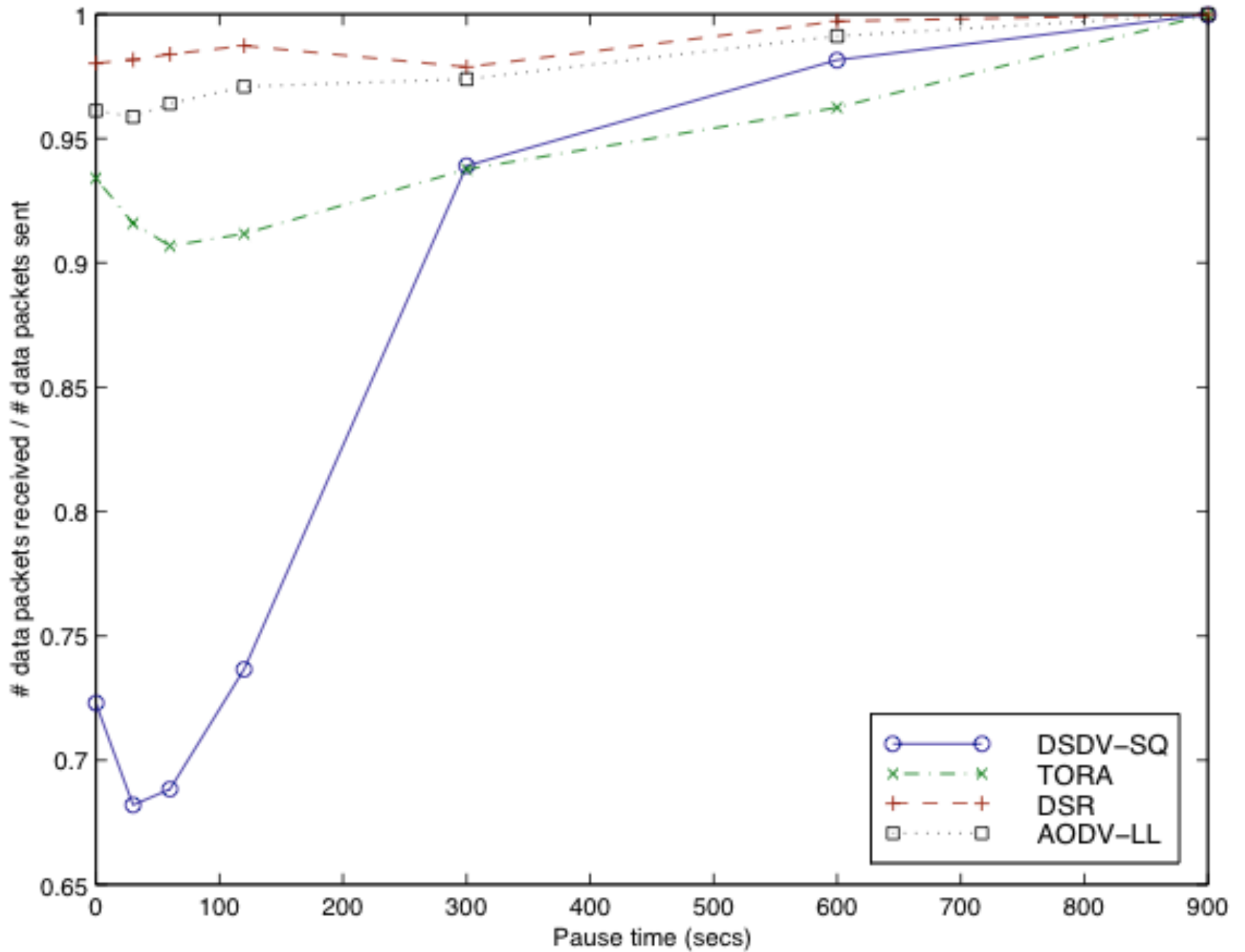
Dynamic Source Routing

- RouteRequest izsūta reaktīvi, kā AODV
- Visu maršrutu sūta līdzī paketē
- Mezgliem-starpniekiem maršruts nav jāglabā

<http://www.dpunkt.de/mobile/code/dsr.html>

Kā ar DSR servisa informācijas  
apjomu (overhead)?





# GPSR, 2000.g

Greedy Perimeter Stateless Routing

- **Greedy:** sūta tam, kas tuvāk mērķim
- **Perimeter:** sasniedzot lokālo maksimumu, atkāpjas pa perimetru
- **Stateless:** neglabā maršrutus, tikai kaimiņu pozīcijas
- **Routing:** (ģeogrāfiskais) maršrutizācijas protokols



# Secinājumi

Kādi ir Jūsu secinājumi?

## 4. Eseja: maršrutizācija

Vai maršrutizācijai ir nepieciešama laika sinhronizācija? Atbildi pamatot!

Termiņš: 26.10.2011. 05:00