Ad hoc Positioning System (APS)

Dragos Niculescu, Badri Nath {dnicules, badri} @cs.rutgers.edu

Rutgers University

summary

- motivation
- GPS review
- APS outline
- APS propagation methods
- simulation results
- conclusions

problem statement

- ad hoc deployed nodes should be able to know their location
  - global coordinates
  - low overhead for mobility
  - accuracy comparable with the node communication range
  - disconnected regions should be able to operate independently
  - without predeployed infrastructure

motivation

- reported information has to be associated with location (sensor networks)
- location helps routing with small or no routing tables
  - geographic routing
  - geodesic routing
  - need global naming
- why not use GPS in each node?
  - battery life
  - form factor
  - line of sight
  - precision

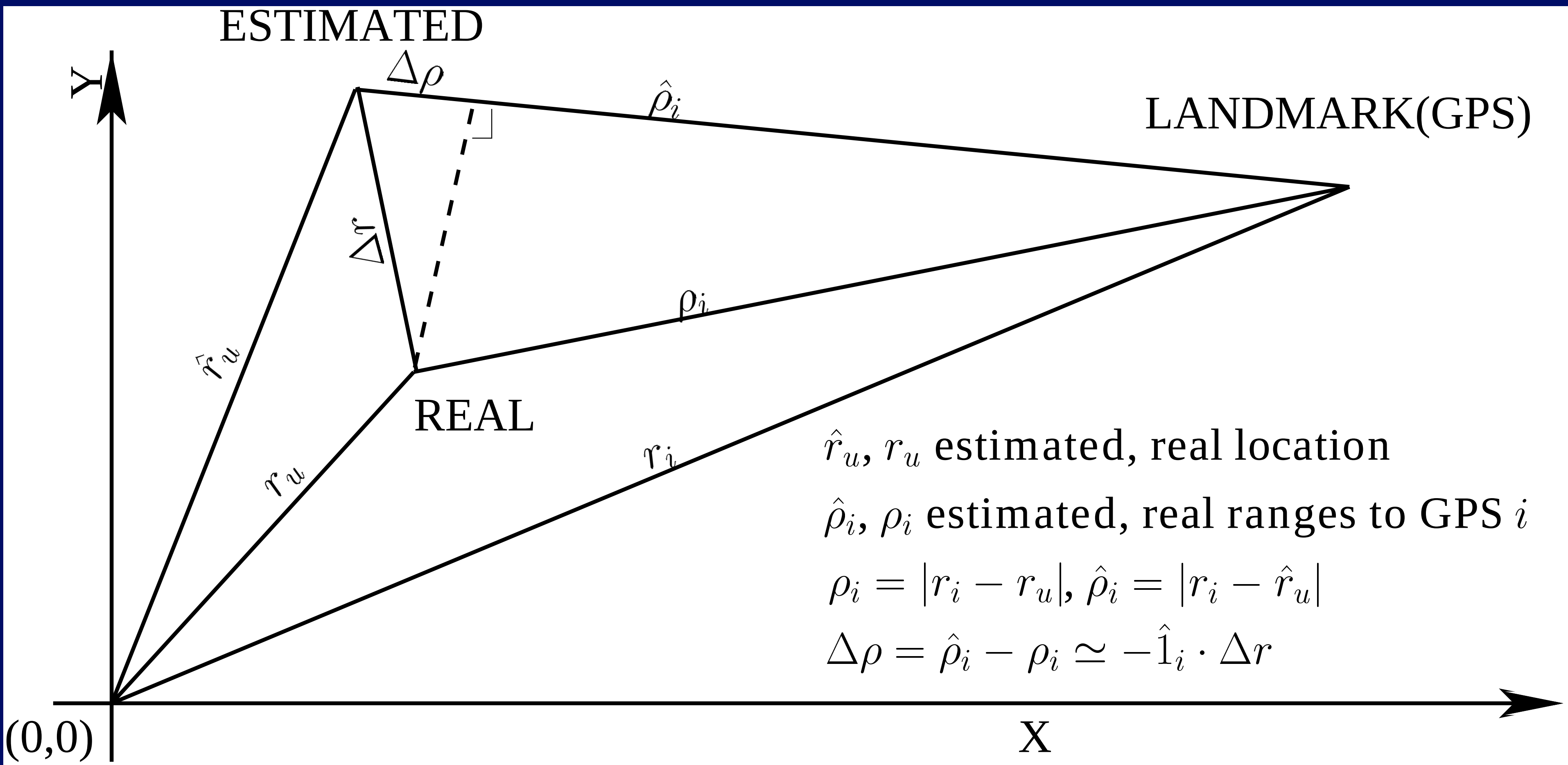
related work

- centralized solution – Berkeley
- positioning using a grid infrastructure – UCLA
- uses radio and ultrasound with ceiling beacons – MIT (CRICKET)
- premaps of the radio properties of the region – Microsoft (RADAR)
- positioning relative to a chosen node – EPFL
- GPS, VOR

GPS review

- Given
  - (imprecise) ranges to at least three satellites, $\hat{\rho}_i$
  - the locations of the satellites $r_i = (x_i, y_i)$
- a node may infer its own location $\hat{r}_u = (x_u, y_u)$
- $(x_i - x_u)^2 + (y_i - y_u)^2 = \hat{\rho}_i^2, i = all\ satellites$
- nonlinear system $\rightarrow$ solved using an iterative method

GPS review


ESTIMATED
Y
$\Delta\rho$
$\hat{\rho}_i$
LANDMARK(GPS)
Δr
ρ_i
$\hat{r}_u$
REAL
r_i
r_u
$\hat{r}_u$, r_u estimated, real location
$\hat{\rho}_i$, ρ_i estimated, real ranges to GPS i
$\rho_i = |r_i - r_u|$, $\hat{\rho}_i = |r_i - \hat{r}_u|$
$\Delta\rho = \hat{\rho}_i - \rho_i \simeq -\hat{1}_i \cdot \Delta r$
(0,0)
X

GPS review

- $\Delta\rho = \hat{\rho}_i - \rho_i \simeq -\hat{1}_i \cdot \Delta r$
- $\hat{1}_i = -\frac{r_i - \hat{r}_u}{|r_i - \hat{r}_u|}$ the unit vector of $\hat{\rho}_i$
- $\Delta r = \hat{r}_u - r_u$ the correction to be applied to the current position
- solve the linear system $\begin{bmatrix} \Delta\rho_1 \\ \Delta\rho_2 \\ \Delta\rho_3 \\ \ldots \\ \Delta\rho_n \end{bmatrix} = \begin{bmatrix} \hat{1}_{1x} & \hat{1}_{1y} \\ \hat{1}_{2x} & \hat{1}_{2y} \\ \hat{1}_{3x} & \hat{1}_{3y} \\ \ldots & \ldots \\ \hat{1}_{nx} & \hat{1}_{ny} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$
- repeat until $\Delta r < \epsilon$

APS outline

- a few nodes (landmarks) know their position
- other nodes infer ranges to at least three non-colinear landmarks
- to estimate distances to neighbors, nodes use
  - signal strength measurement
  - hop count
- a hybrid between GPS and distance vector routing
  - like in DV, distances to landmarks are propagated hop by hop
  - like in GPS, each node estimates its own location
- each landmark is treated independently at each node
- may use different methods to propagate distance

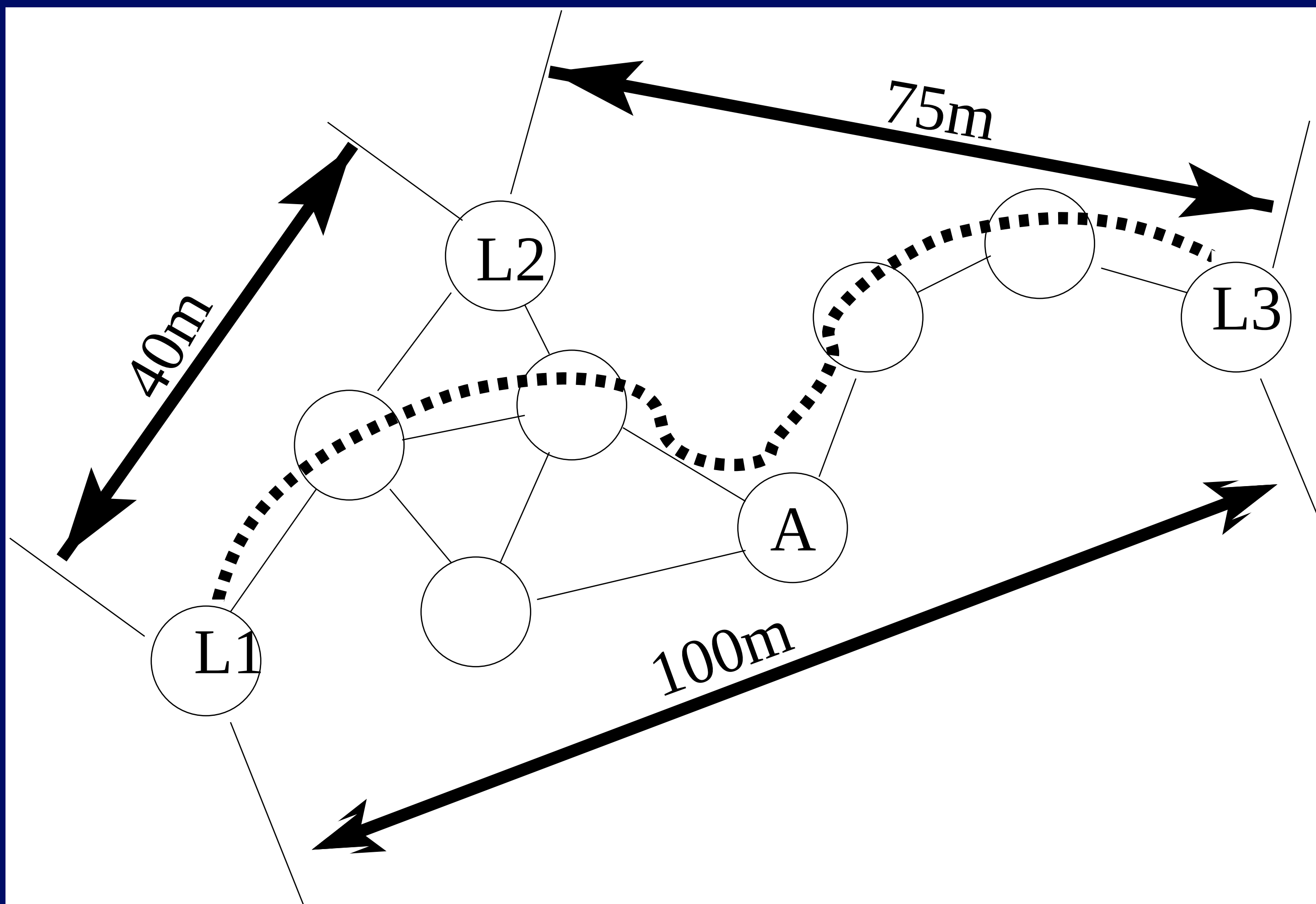
APS - distance propagation

- like in DV, neighbors exchange estimate distances to landmarks
- four possible propagation methods
  - *"DV-hop"* - distance to landmark, in hops (this is standard DV)
  - *"DV-distance"* - travel distance, in meters
  - *"Euclidean"* - euclidean distance to landmark
  - *"Coordinate"* - node's own coordinates in landmark's coordinate system

“*dv-hop*” propagation

- standard DV propagation
- never measures the distance between neighbors $\rightarrow$ insensitive to SS errors
- each node maintains a table $\{X_i, Y_i, h_i\}$ by running classic DV
- each landmark $\{X_i, Y_i\}$
  - computes a correction $c_i = \frac{\sum \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}}{\sum h_i}, i \neq j$
  - ...and floods it into the network
- each node
  - uses the correction from the closest landmark
  - multiply its hop distances by the correction

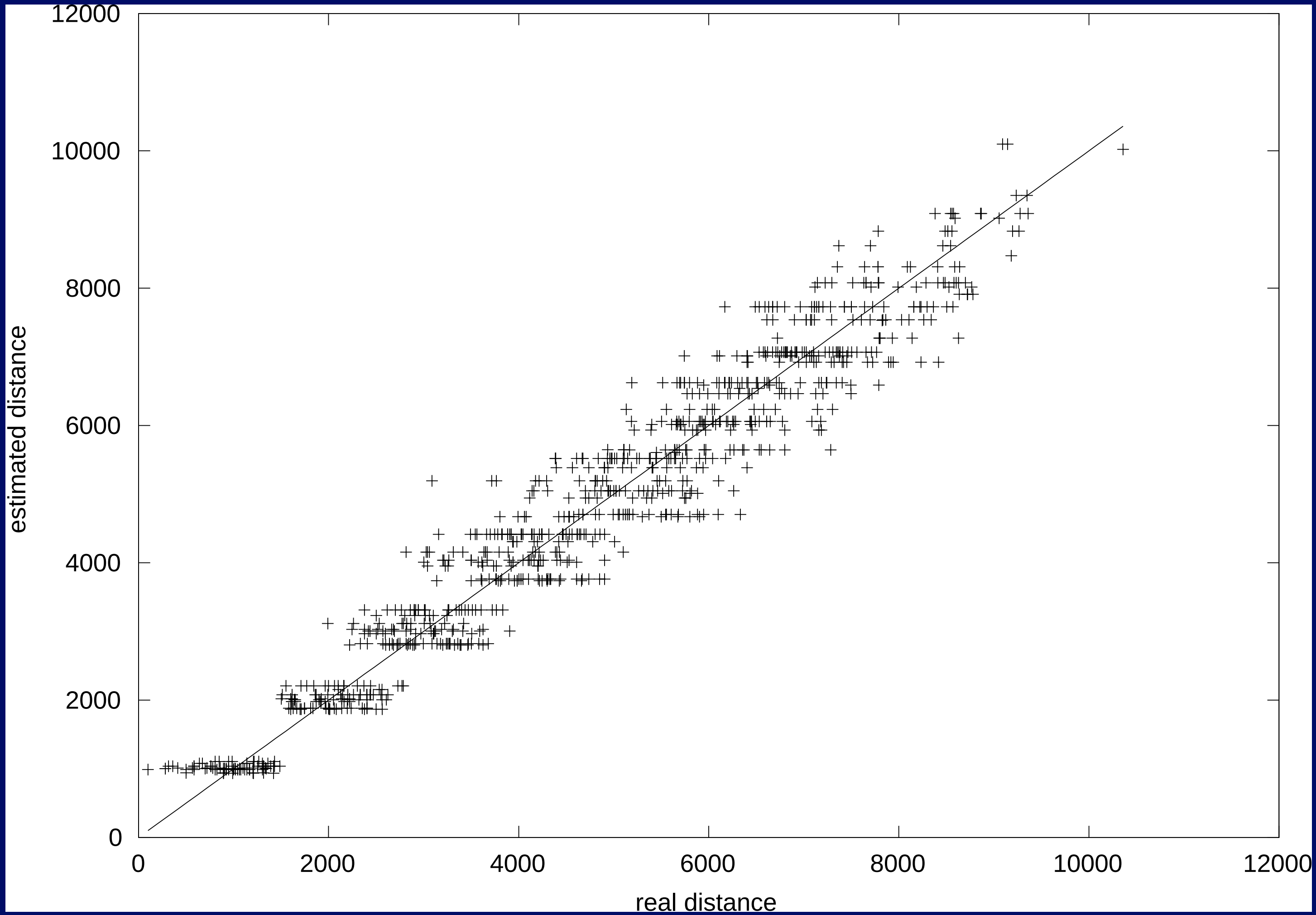
“*dv-hop*” propagation - example



“*dv-hop*” propagation - example

- corrections computed by the landmarks
  - $L_1 \rightarrow \frac{100+40}{6+2} = 17.5$
  - $L_2 \rightarrow \frac{40+75}{2+5} = 16.42$
  - $L_3 \rightarrow \frac{75+100}{6+5} = 15.90$
- assume A gets its correction from L_2
- its estimate distances(ranges) to the three landmarks would be
  - to $L_1 \rightarrow 3 \cdot 16.42$
  - to $L_2 \rightarrow 2 \cdot 16.42$
  - to $L3 \rightarrow 3 \cdot 16.42$
- A performs GPS triangulation with the above ranges

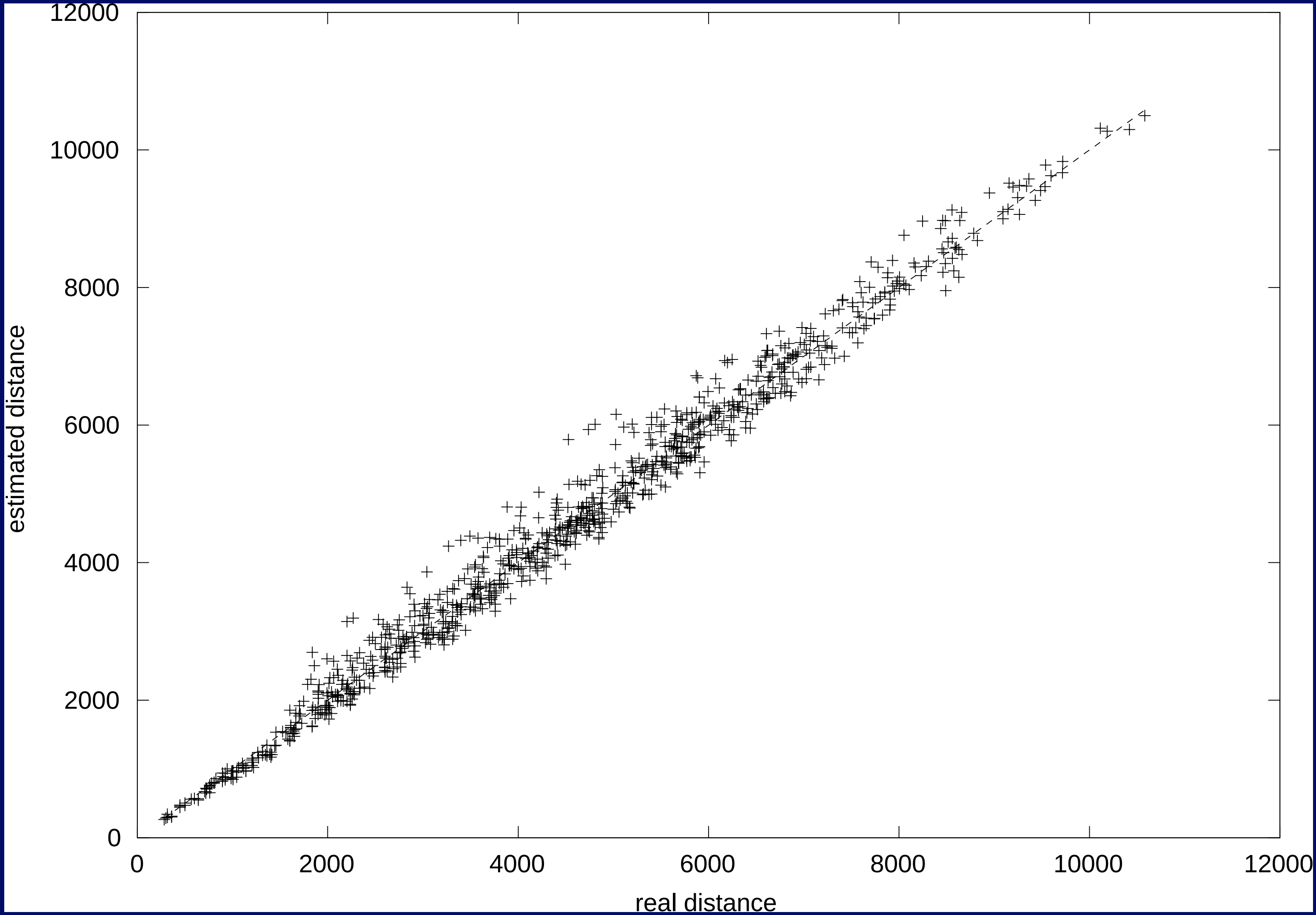
“*dv-hop*” propagation



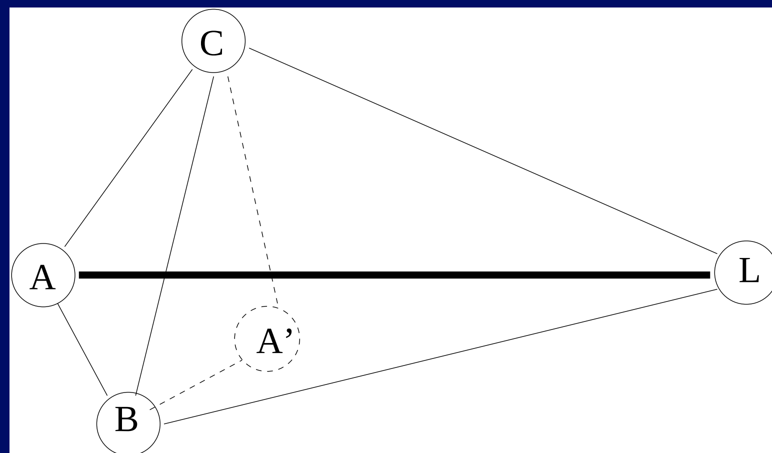
"*dv-distance*" propagation

- DV propagation using travel distance, in meters
- each <u>node</u> maintains a table $\{X_i, Y_i, d_i\}$
- each <u>landmark</u> $\{X_i, Y_i\}$
  - computes a correction $c_i = \frac{\sum \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}}{\sum d_i}, \; i \neq j$
  - ...and floods it to its neighbors
- each node
  - uses the correction from the closest landmark
  - multiply its distances by the correction

“*dv-distance*” propagation

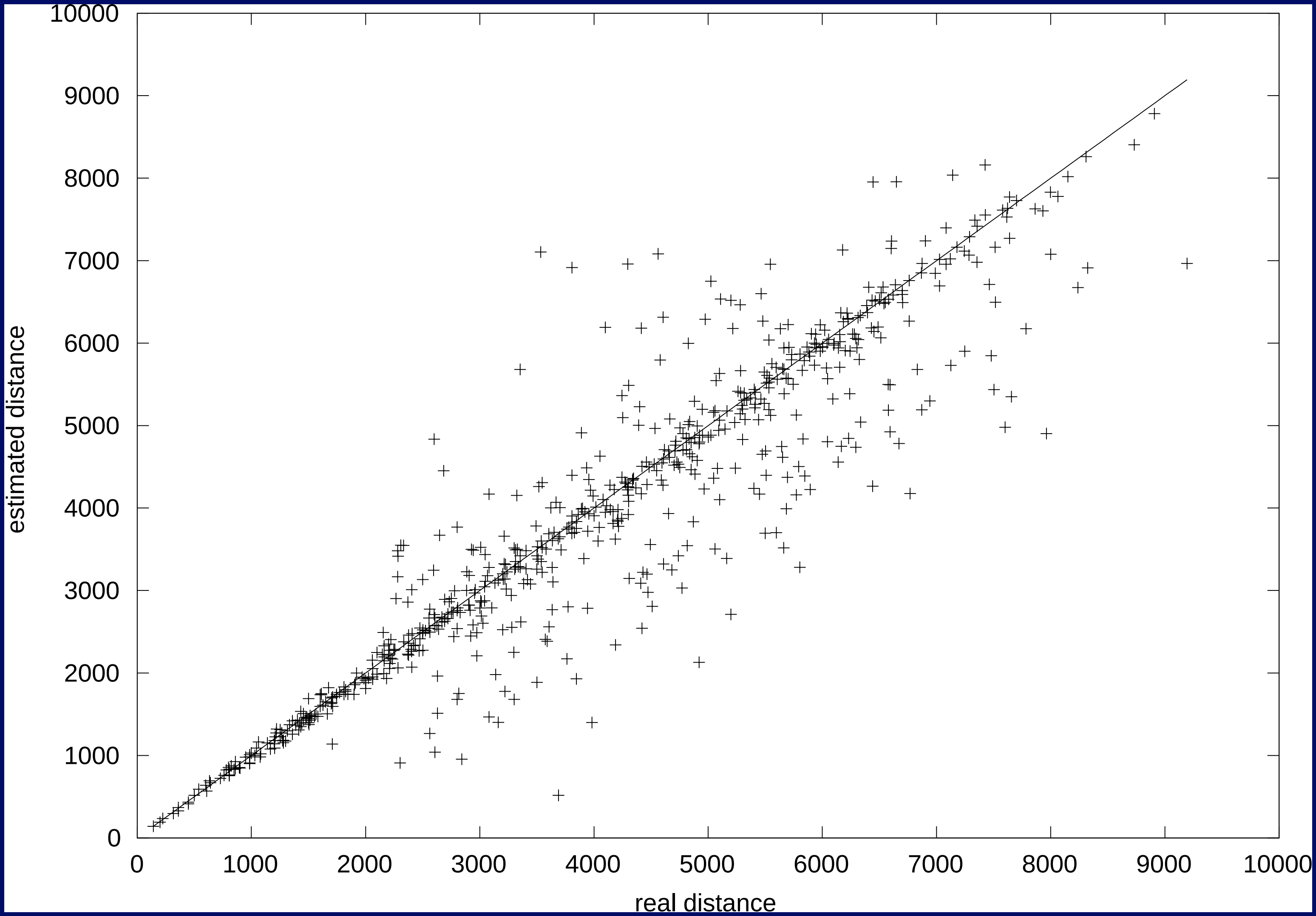


“*euclidean*” propagation

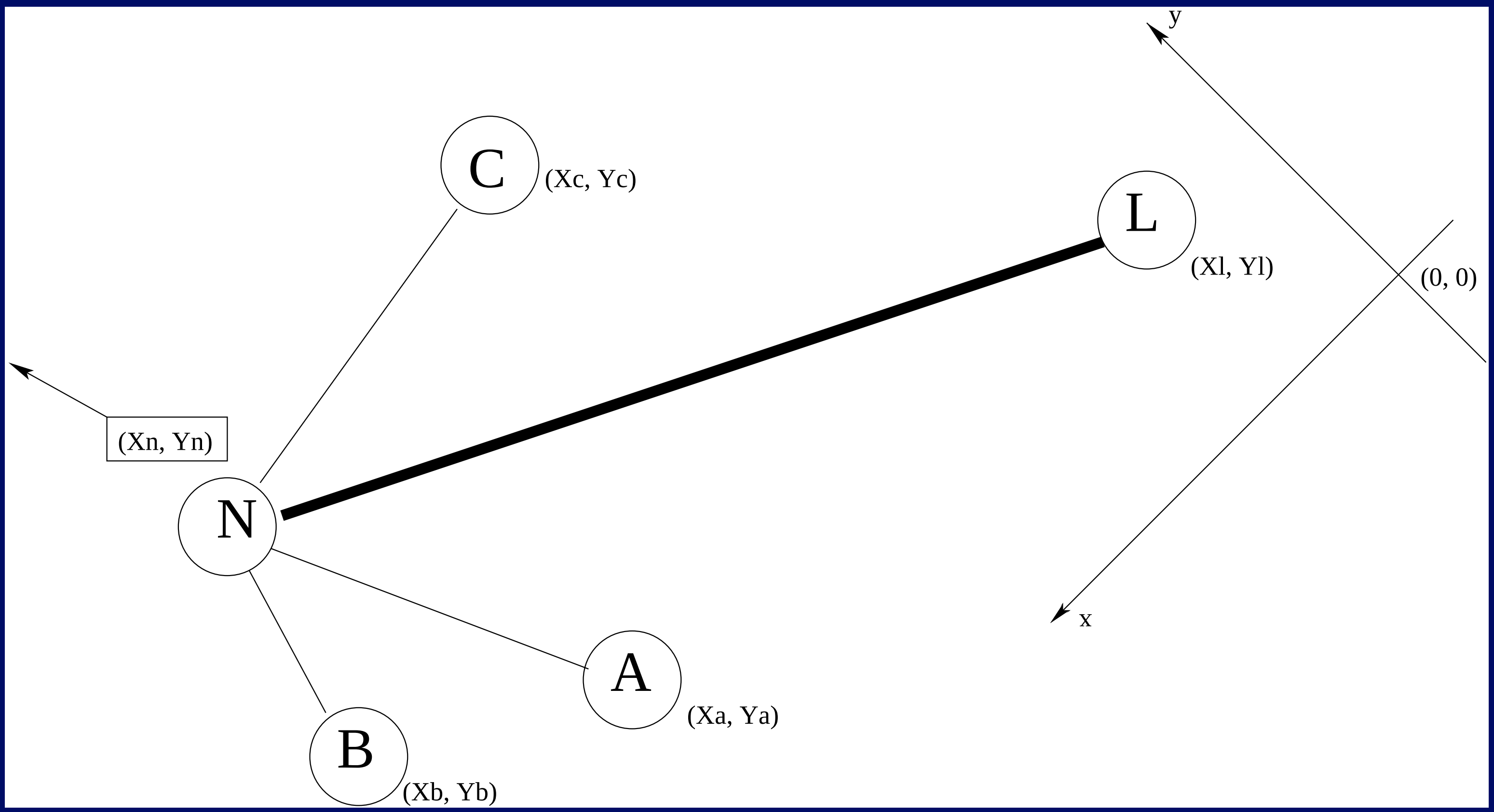


- node A
  - measures distances to immediate neighbors B and C
  - learns distance BC from either B or C,
  - or, possibly infers it by mapping all its neighbors
- B and C know their euclidean distances to landmark L
- A has to find the diagonal AL

“*euclidean*” **propagation**



"*coordinate*" propagation



- each landmark i chooses a random coordinate system in which its coordinates are the true (X_{Li}, Y_{Li}), obtained from GPS

“*coordinate*” propagation

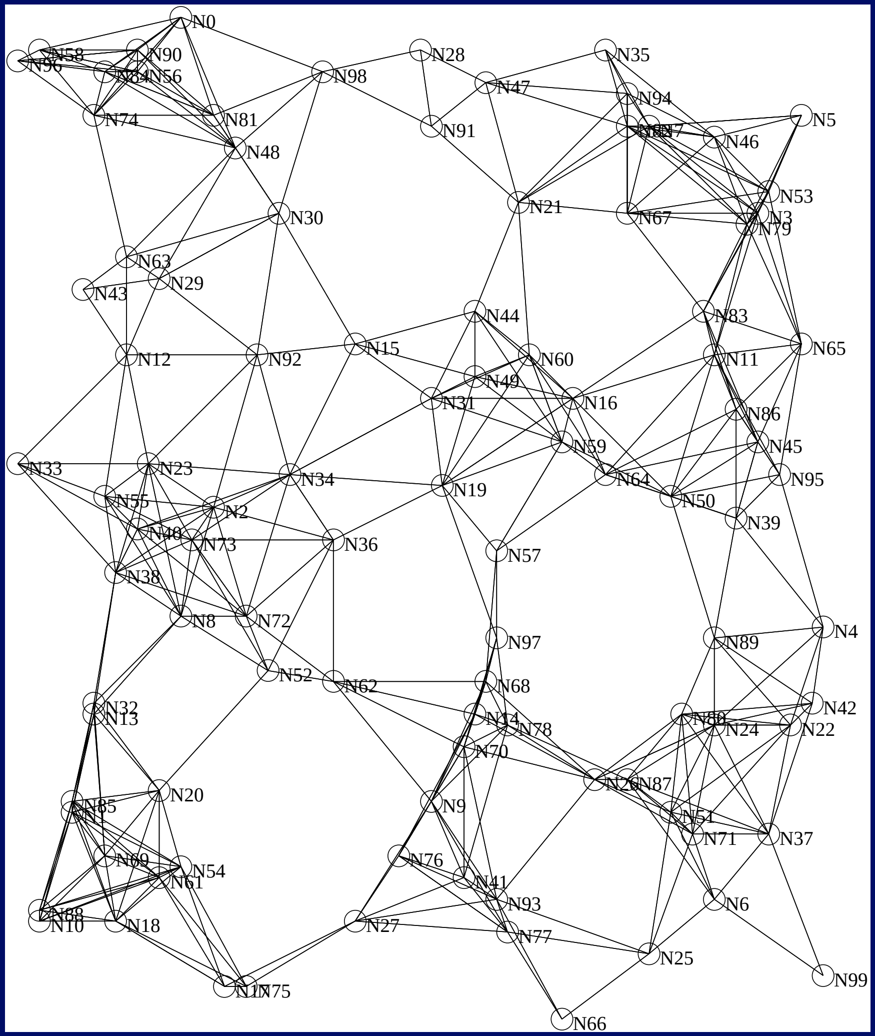
- a node N
  - maintains a table $\{(X_i, Y_i), (X_{Li}, Y_{Li})\}$
  - measures distances to neighboring nodes
  - when having the coordinates of three neighbors, can compute its own coordinates (X_n, Y_n) using the same GPS procedure
- signaling is 50% more than the euclidean method (sends (X_n, Y_n) instead of d_n)
- both *Euclidean* and *Coordinate* methods need second hop information

simulation

- ns-2 based
- random topologies 100-300 nodes
  - isotropic[1] √
  - anisotropic
    - connectivity √
    - radio range ×
    - density ×
- performance metrics
  - absolute location error √
  - geodesic routing overhead √
  - messaging complexity √

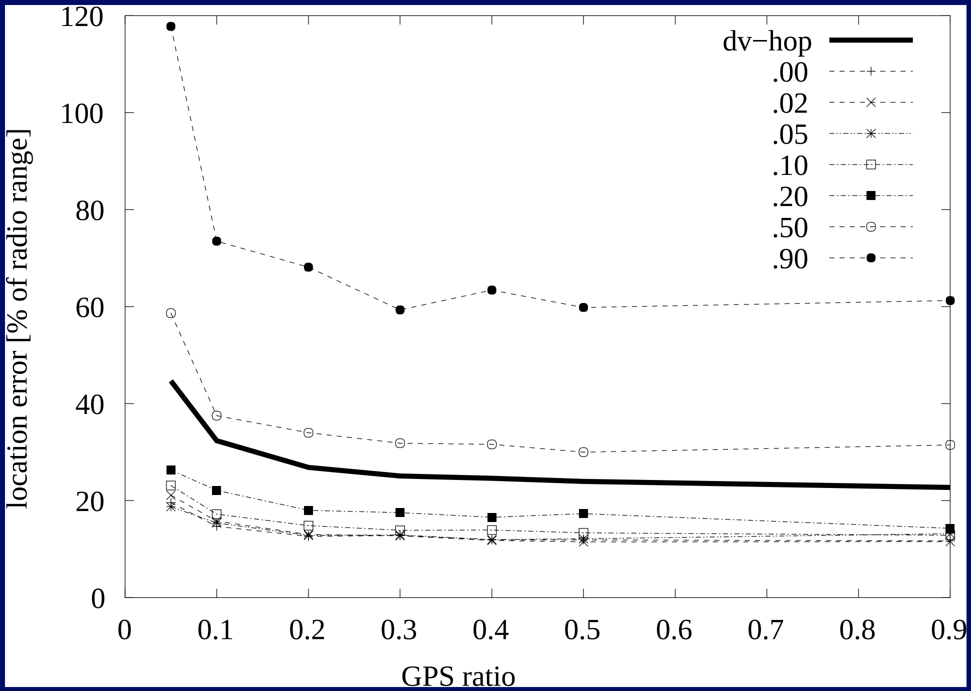
[1] the network has the same properties (density, radio range) in all directions

location error - isotropic

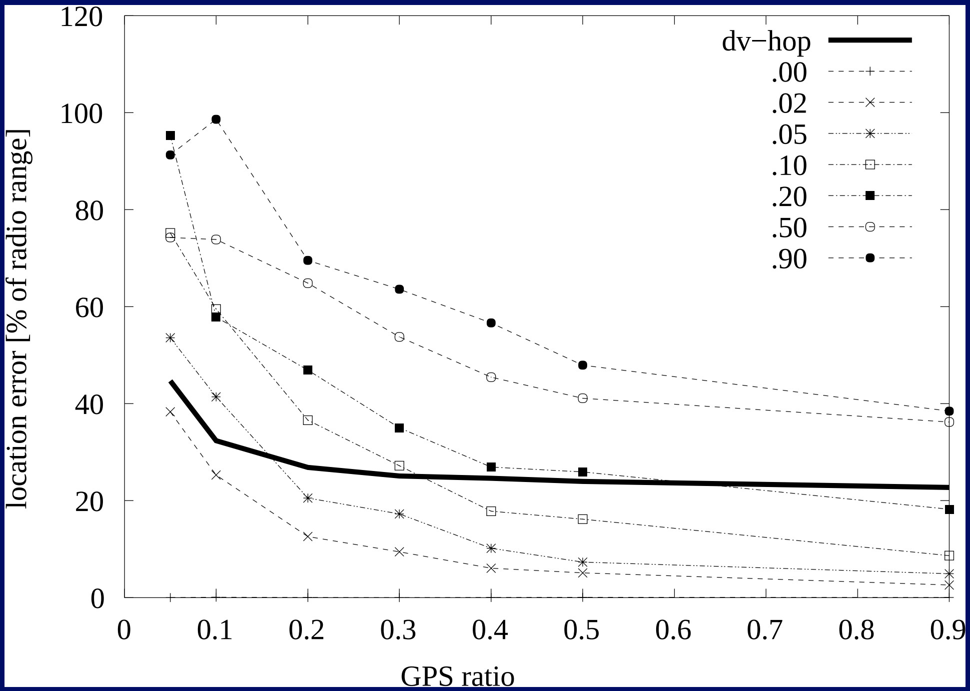


location error - isotropic

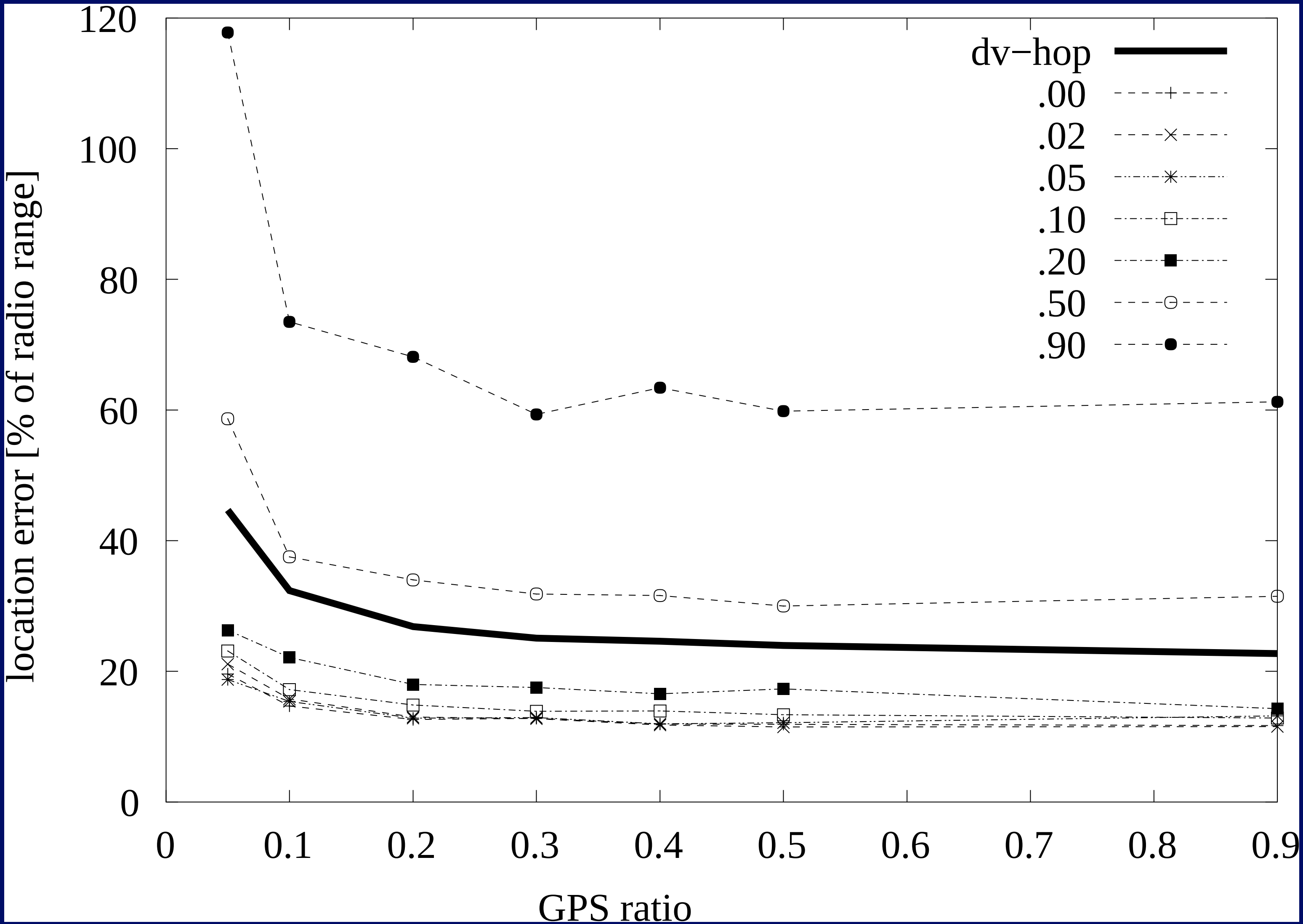
DV-distance



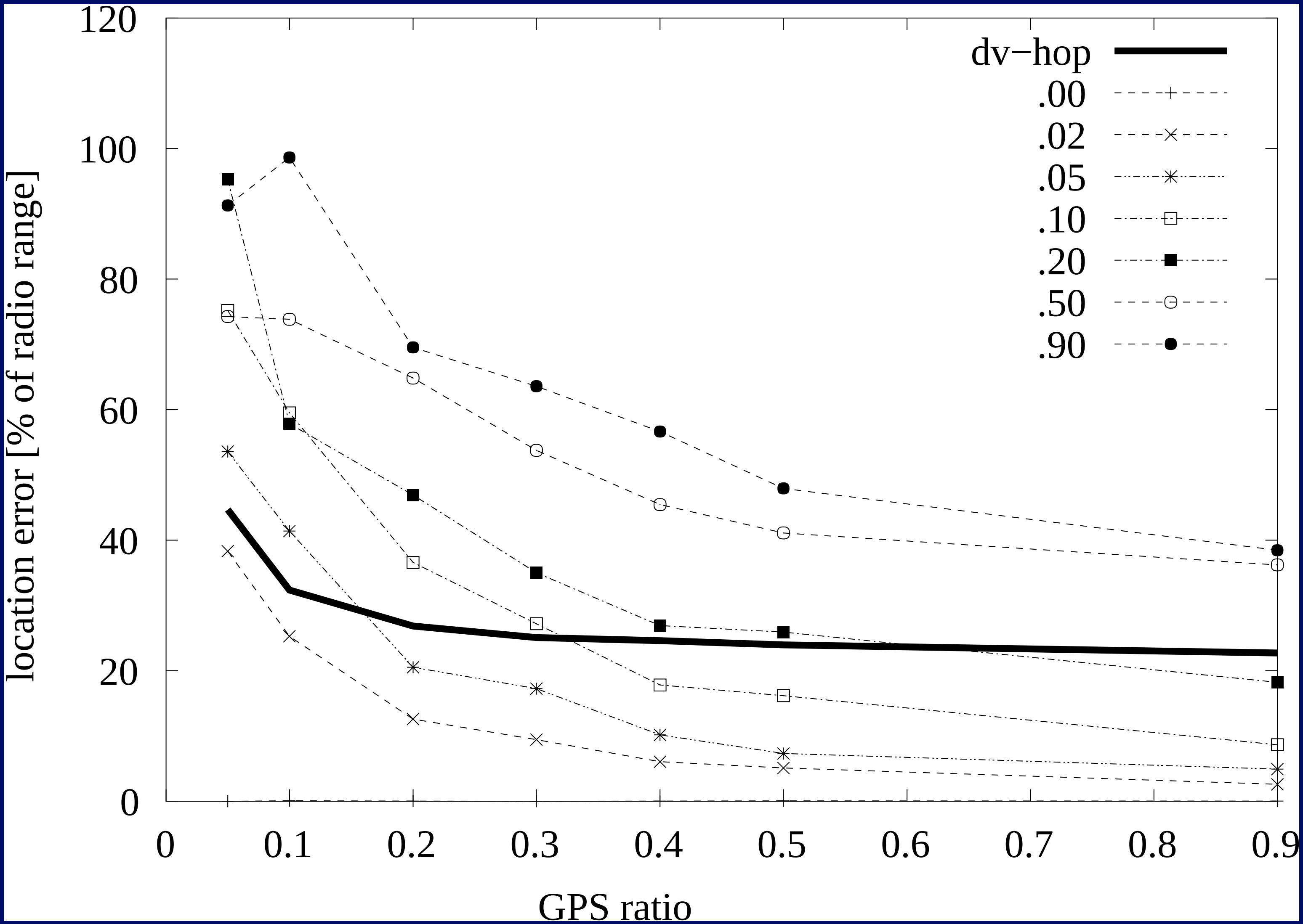
Euclidean



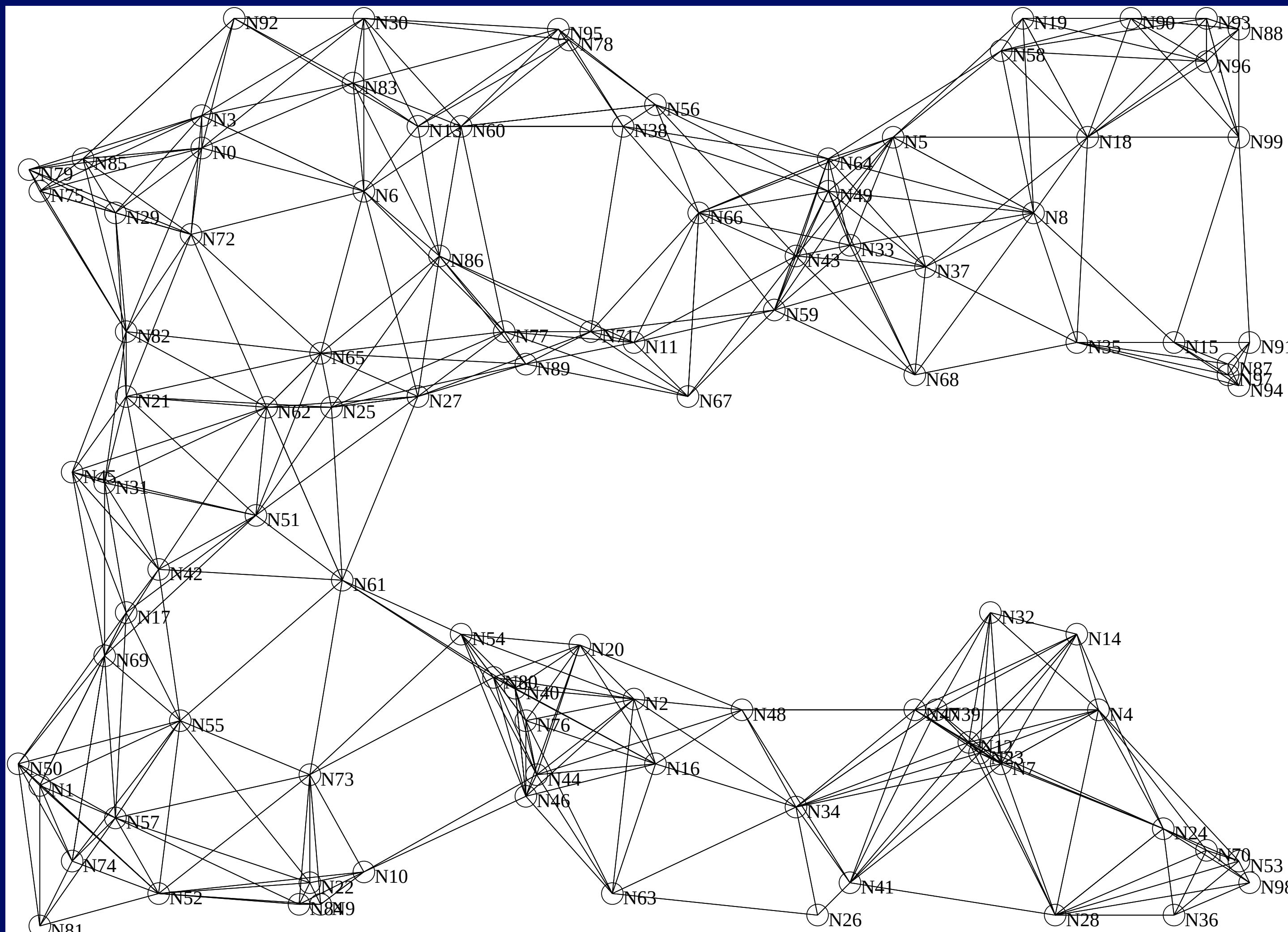
location error - isotropic - *DV-distance*



location error - isotropic - *Euclidean*

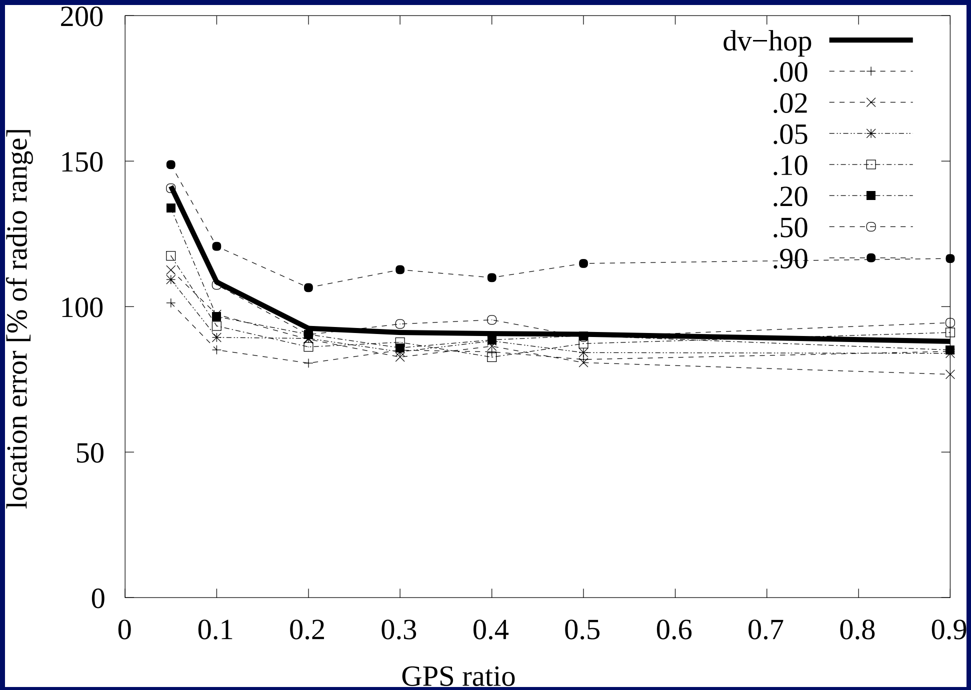


location error - anisotropic

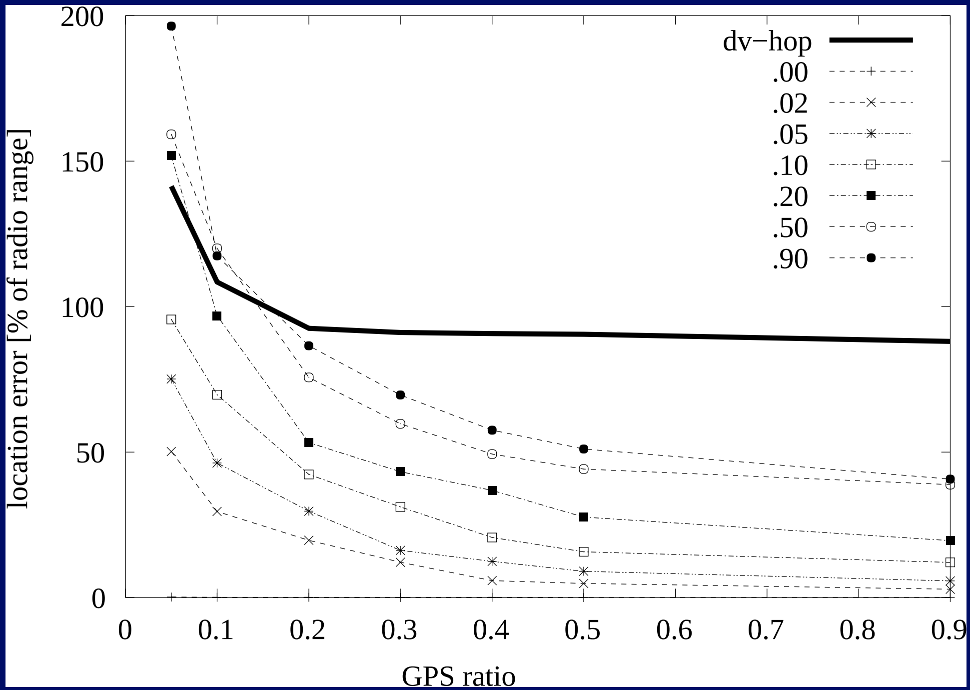


location error - anisotropic

DV-distance

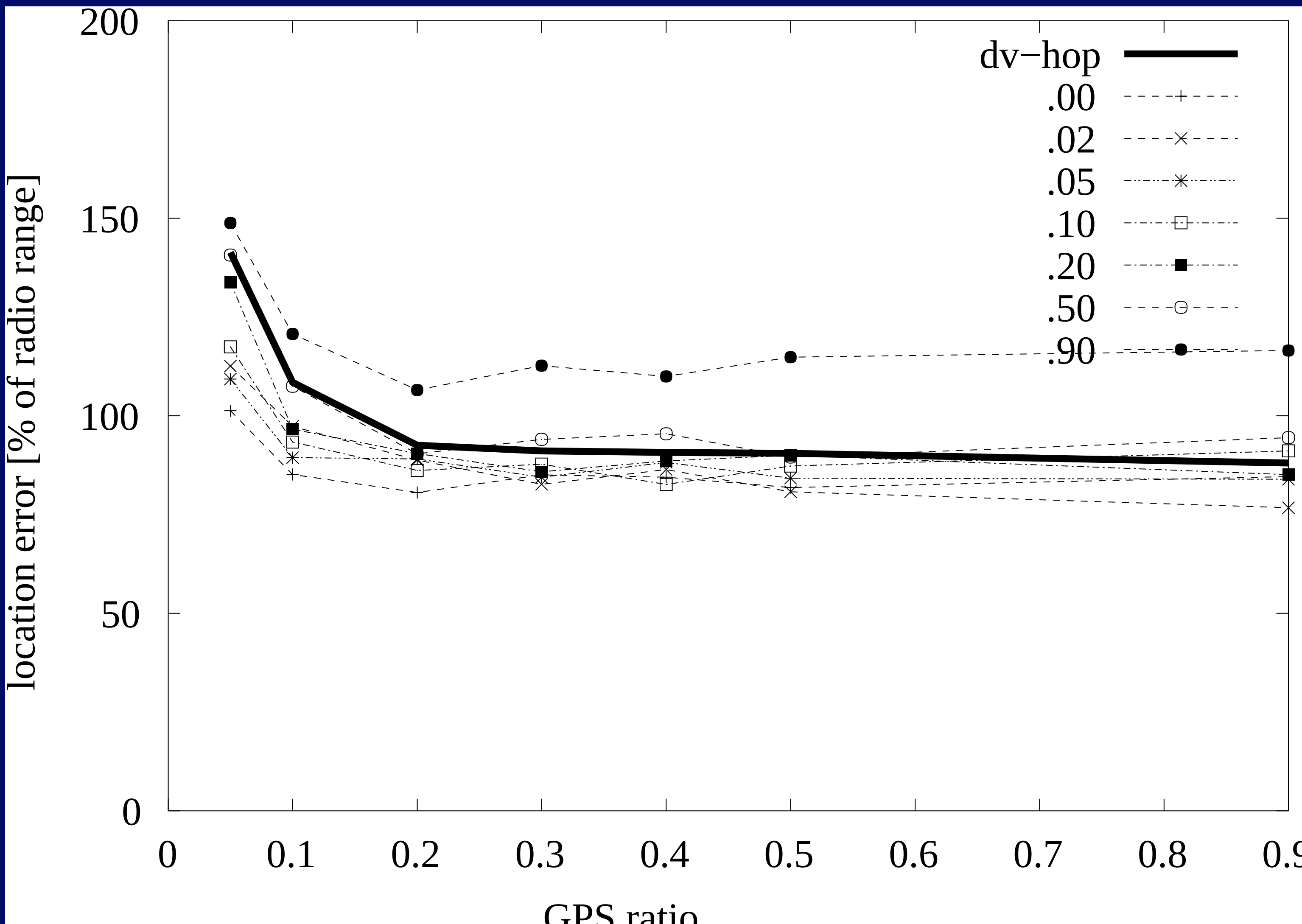


Euclidean

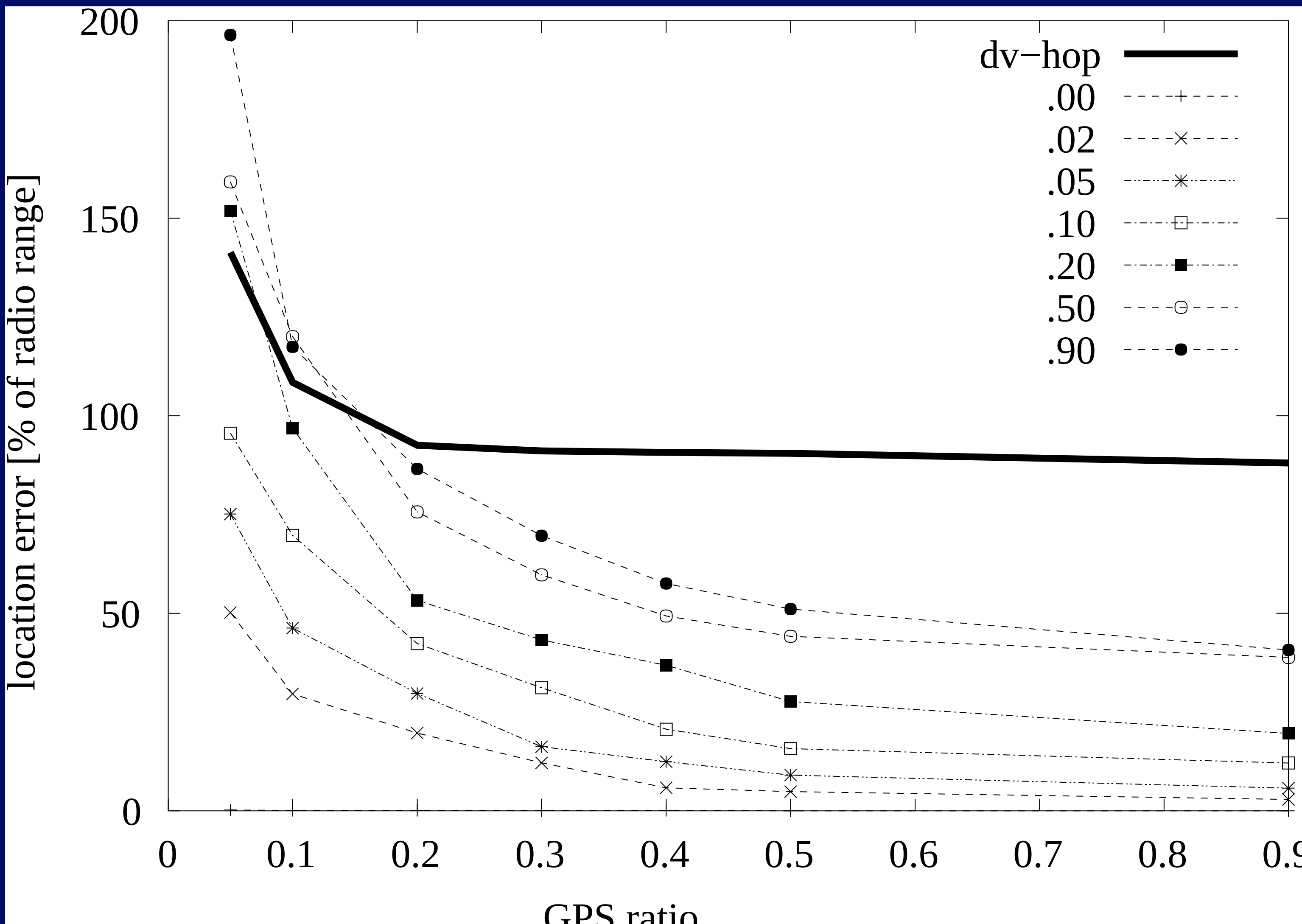


- little variance for “*Euclidean*” across topologies
- anisotropy caused error matters more than measurement error

location error - anisotropic - *DV-distance*

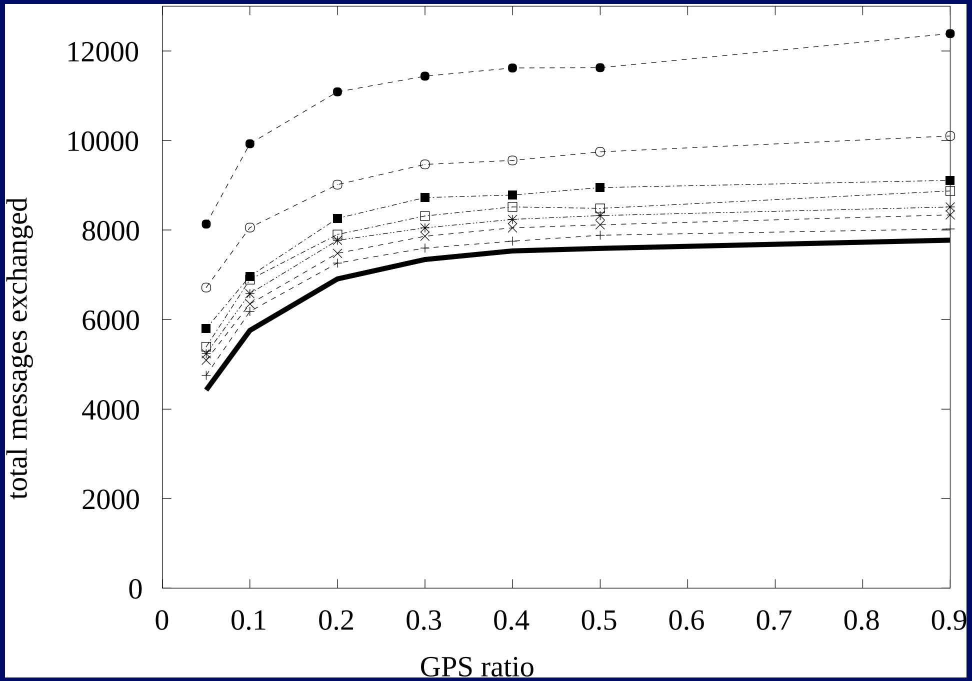


location error - anisotropic - *Euclidean*

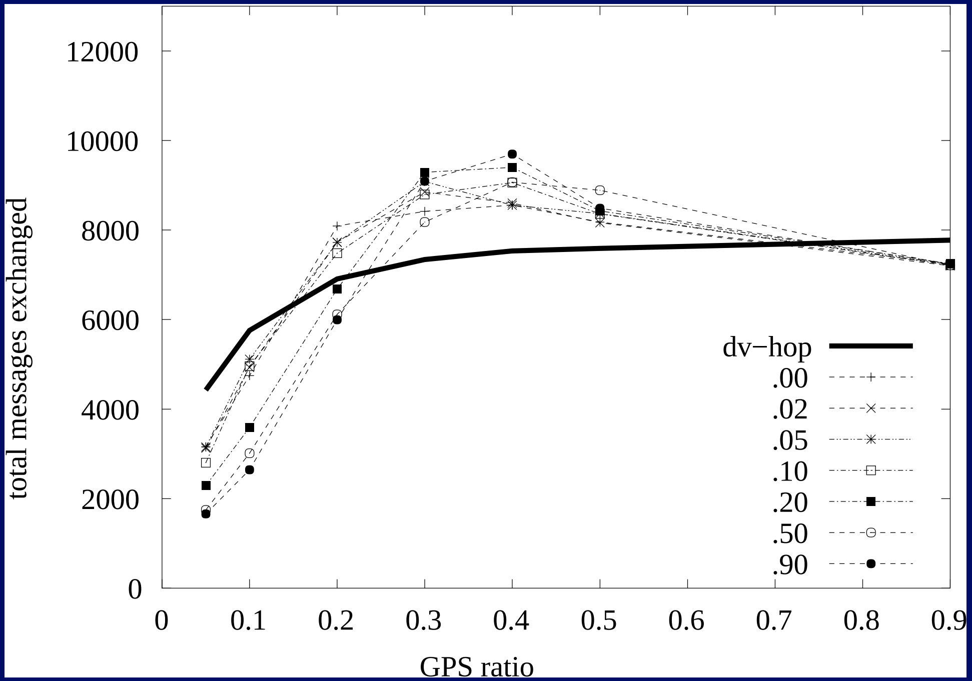


number of messages exchanged

DV-distance

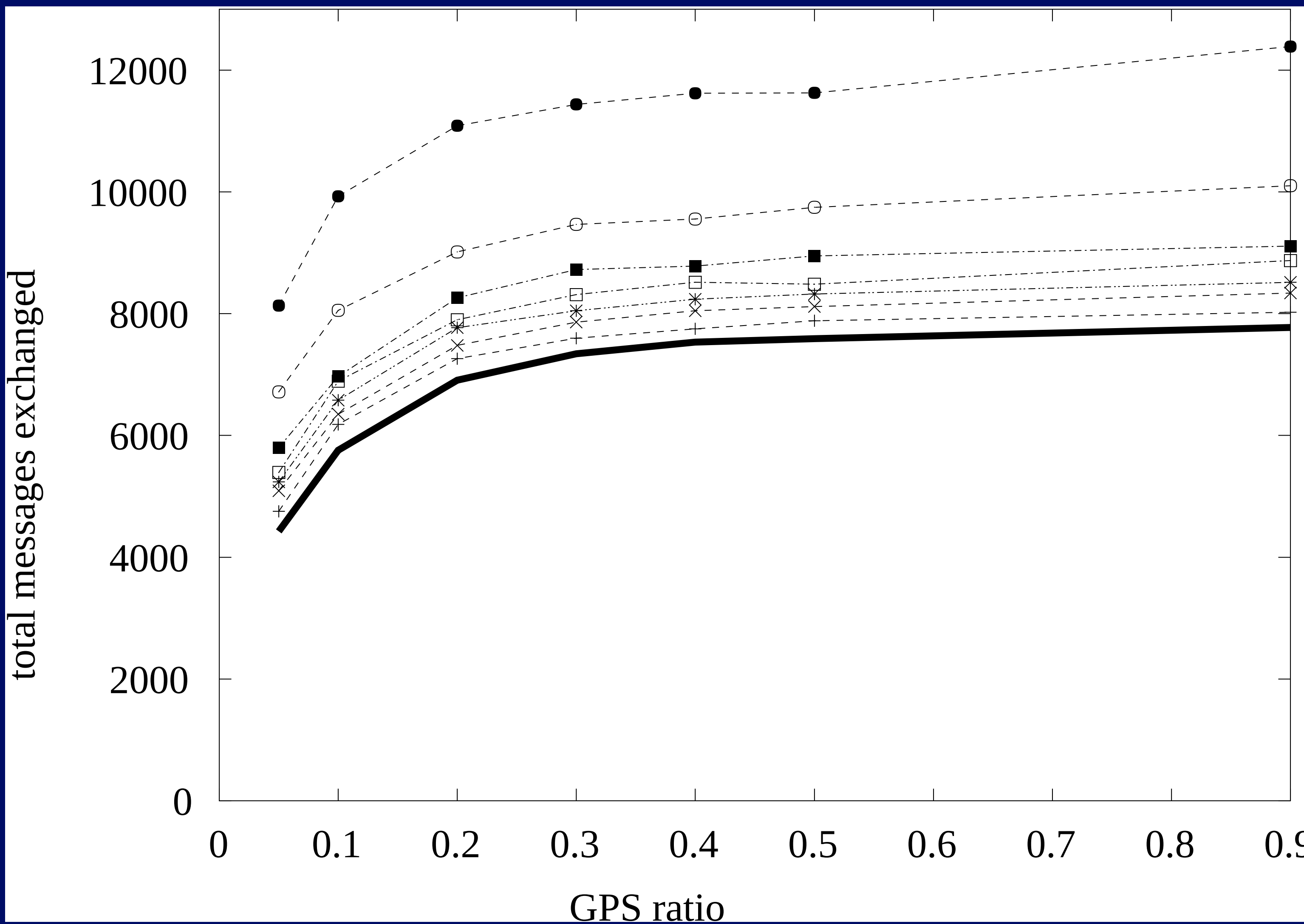


Euclidean

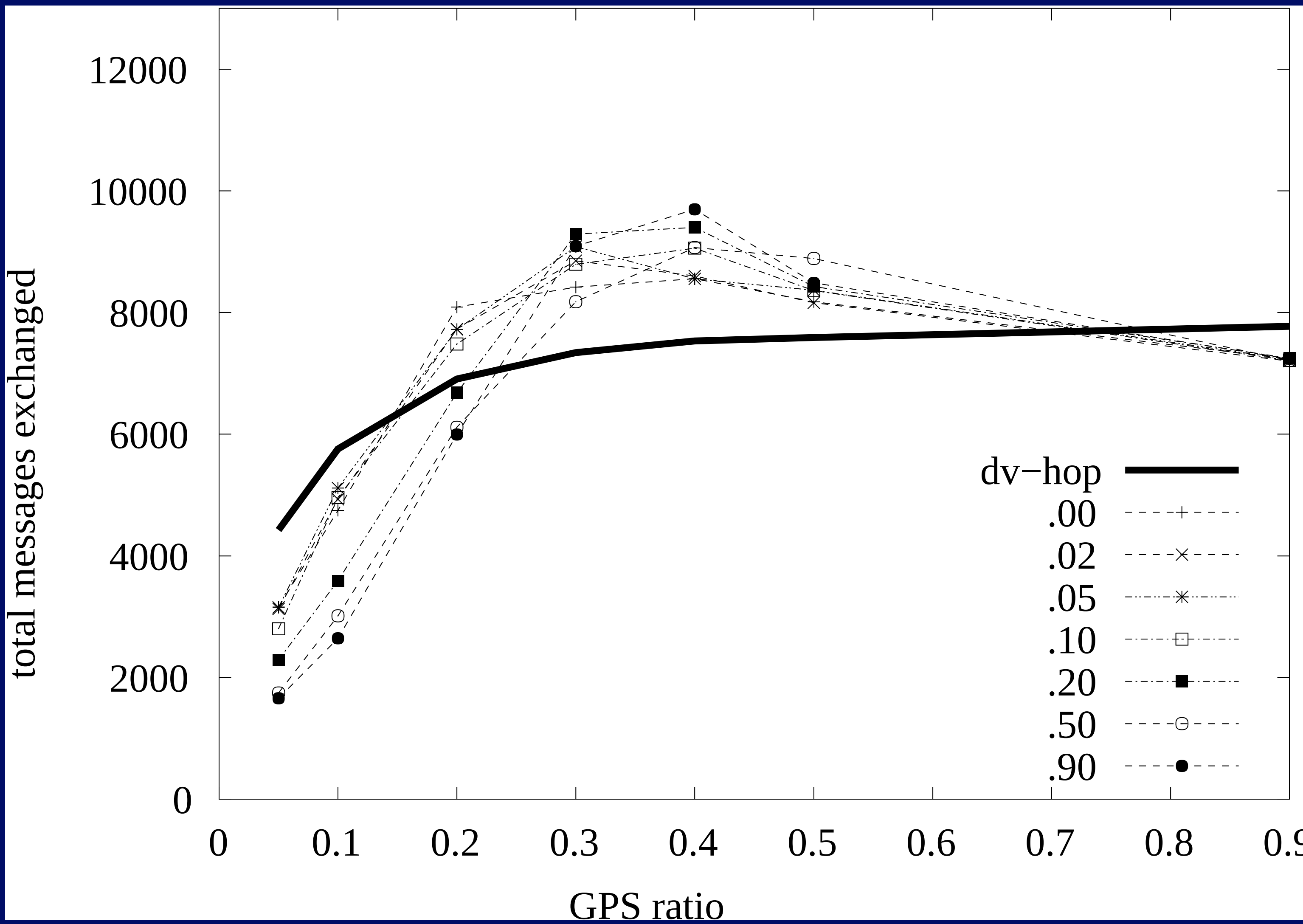


- *DV-distance* updates the same path several times under high error

messages exchanged - *DV-distance*

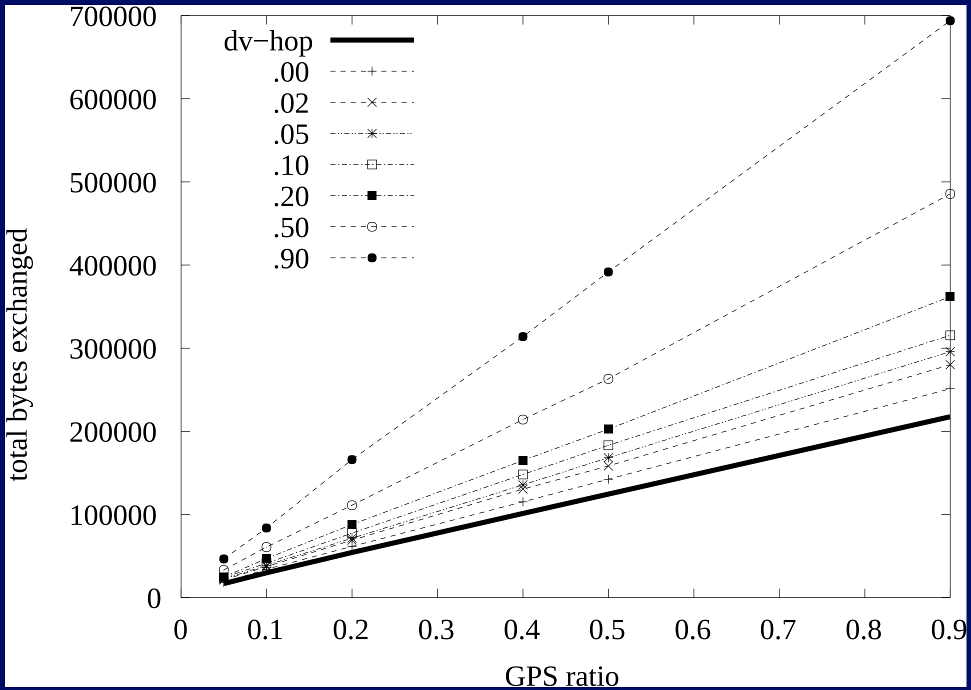


messages exchanged -*Euclidean*

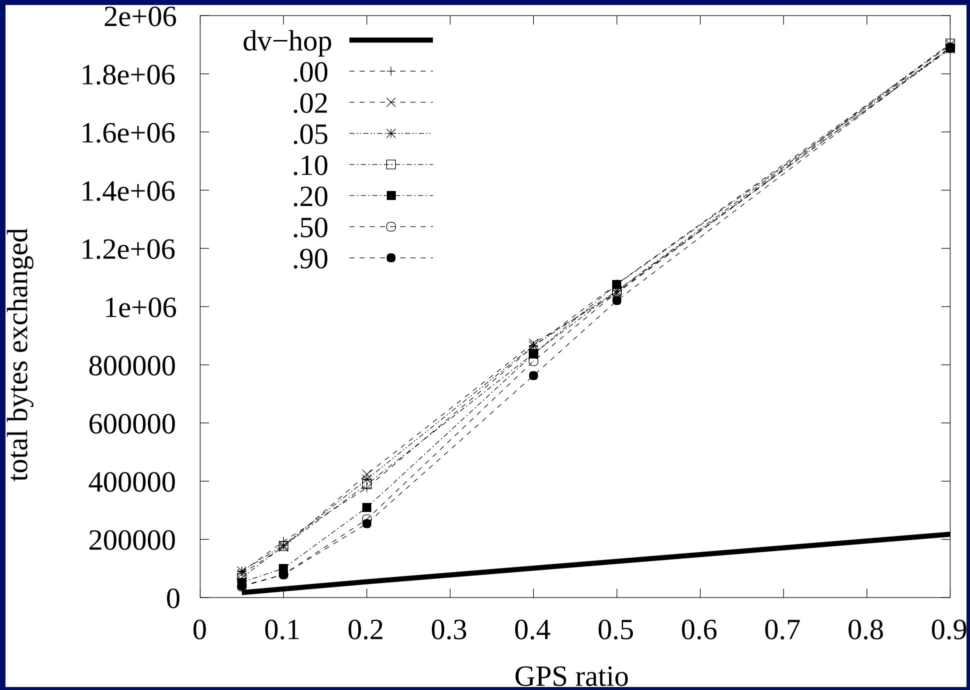


number of bytes exchanged

DV-distance

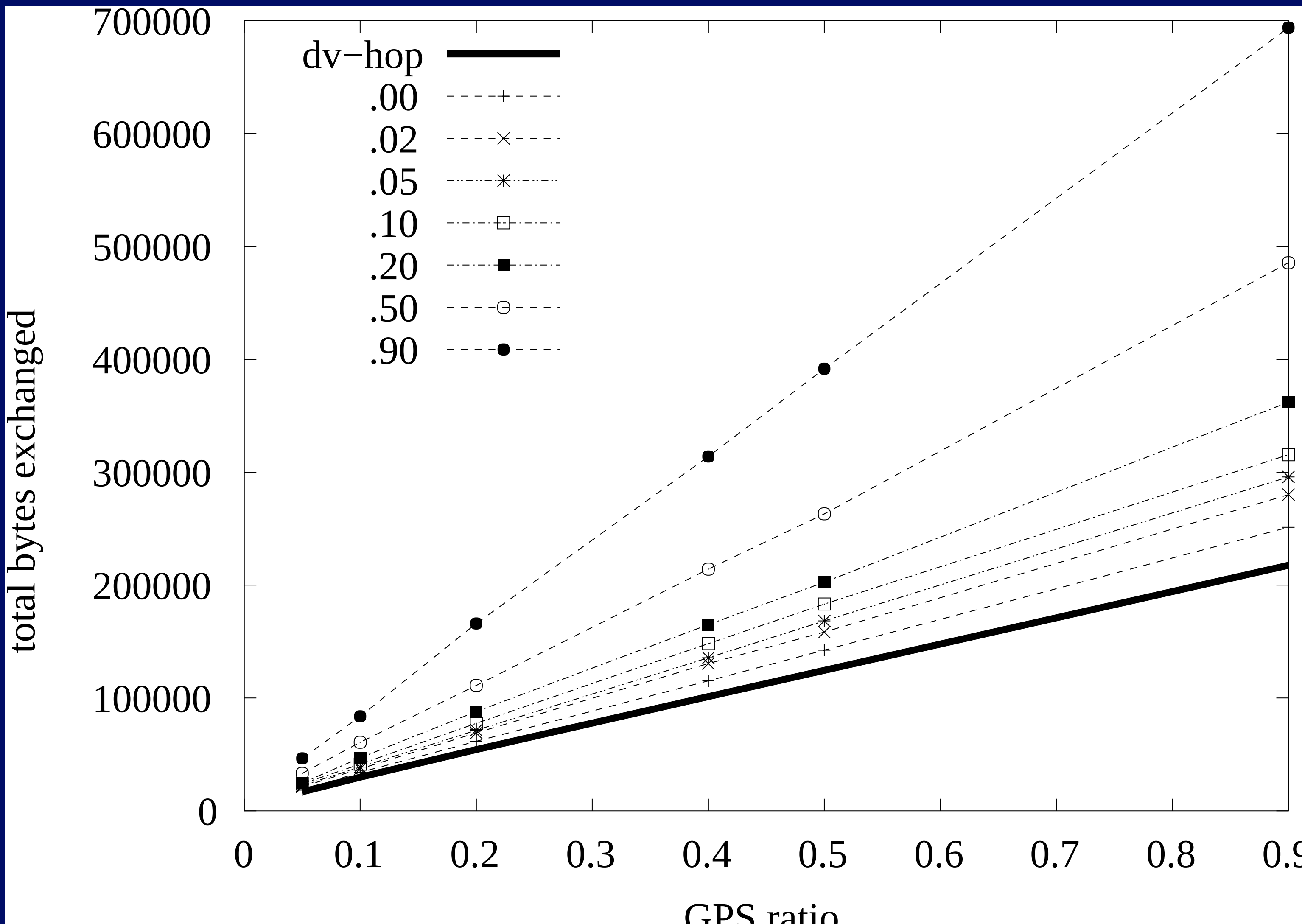


Euclidean

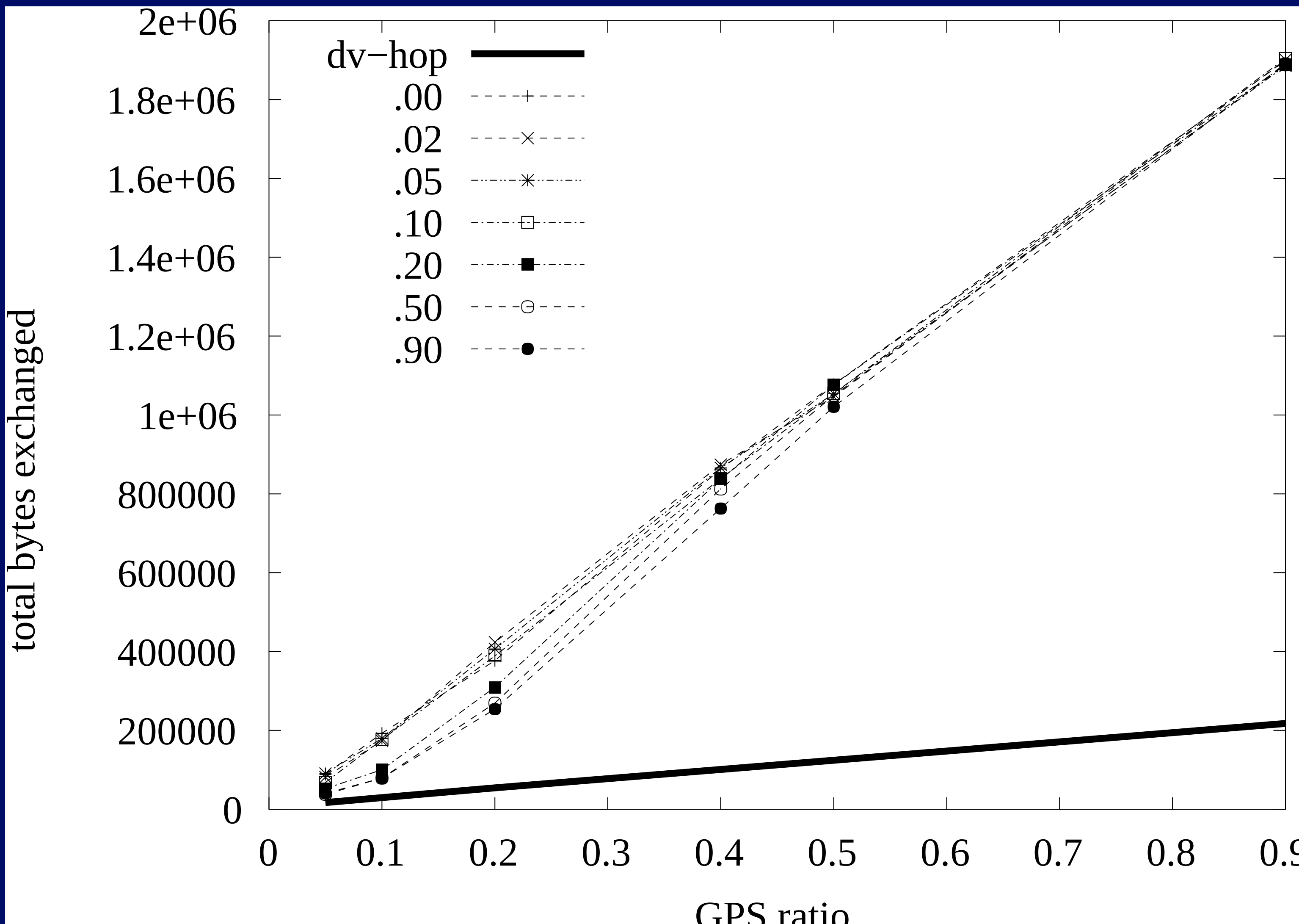


- amount of data exchanged depends on the degree of the graph
- *Euclidean* needs second hop information → higher degree

bytes exchanged - *DV-distance*



bytes exchanged - *Euclidean*

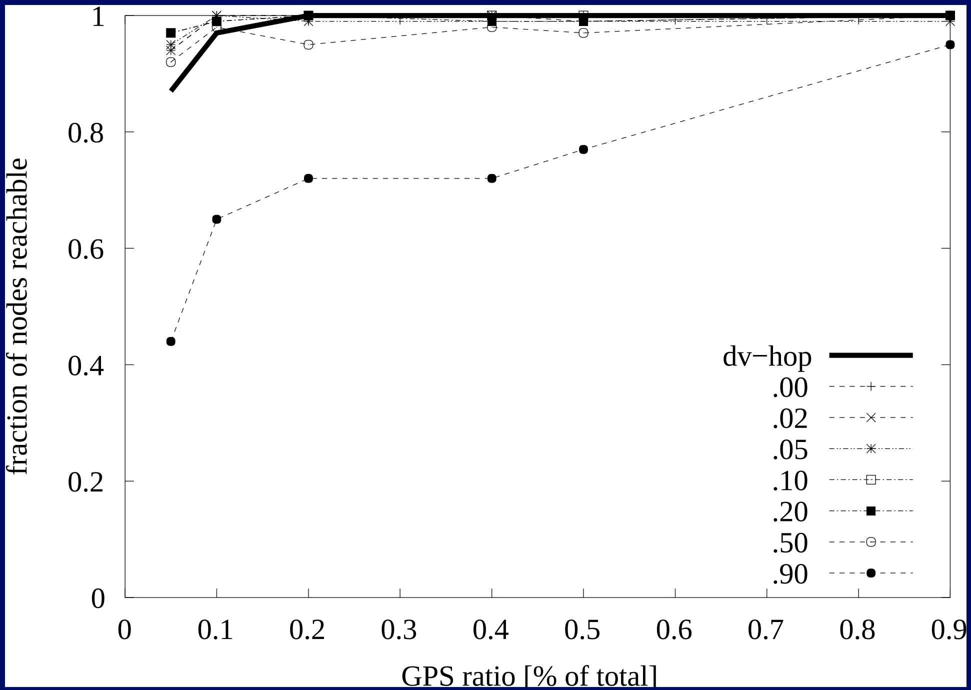


geodesic routing

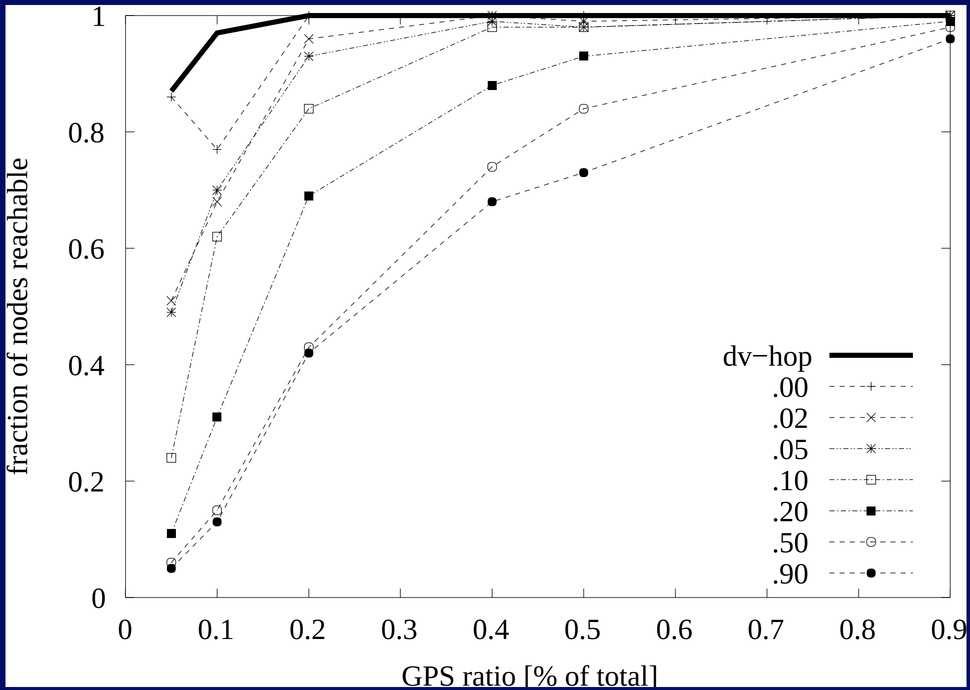
- simple, greedy forwarding decision
  - choose the next hop that is closest to destination
  - closest = in euclidean distance
- no routing loops $\rightarrow$ distance to destination monotonically decreases
- packets may be dropped
  - due to location aberrations
  - intermediate nodes without a computed location
  - destination without a computed location
  - cannot route around obstacles
- can we use geodesic routing with estimated locations?

geodesic routing - reachability

DV-distance

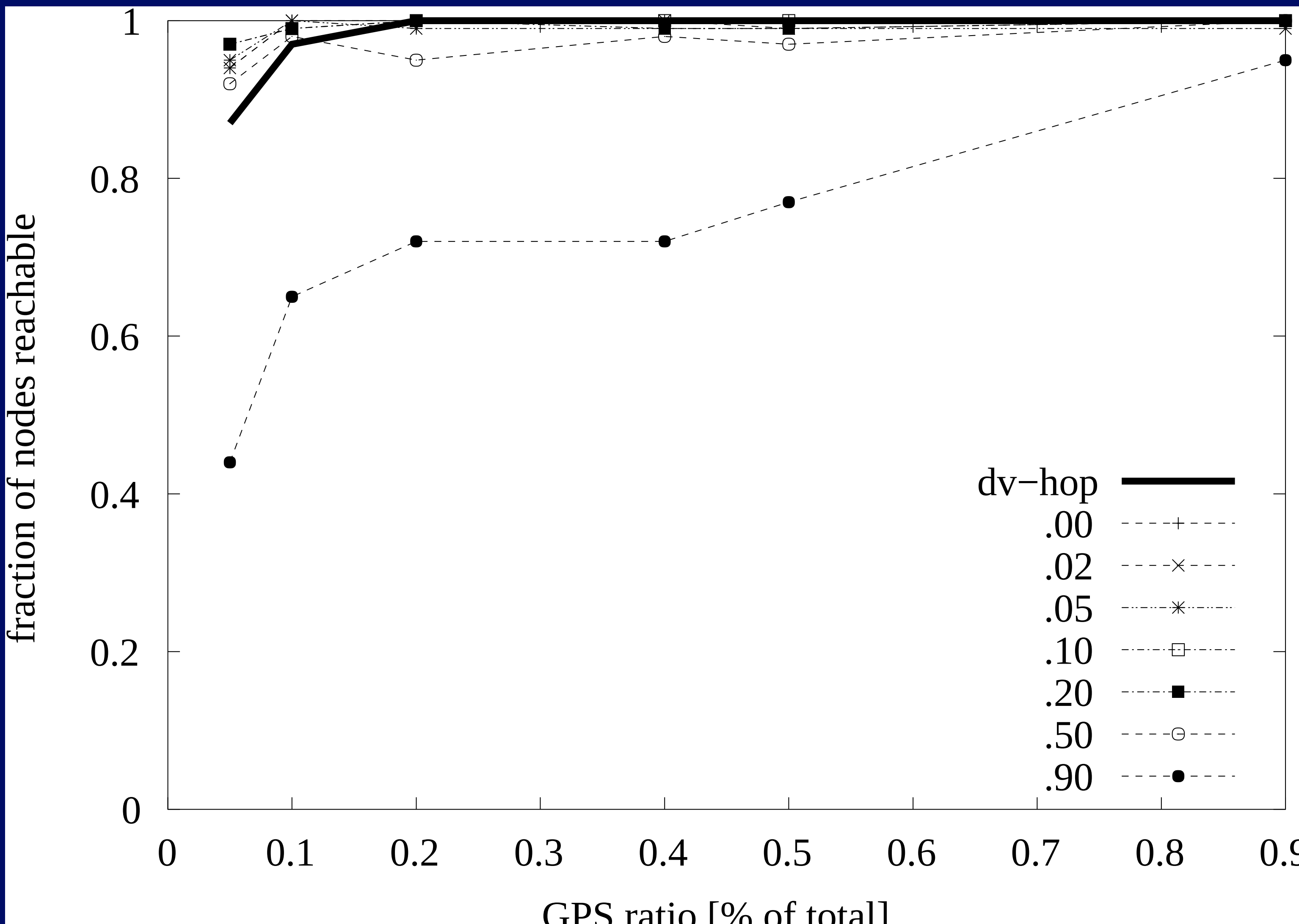


Euclidean

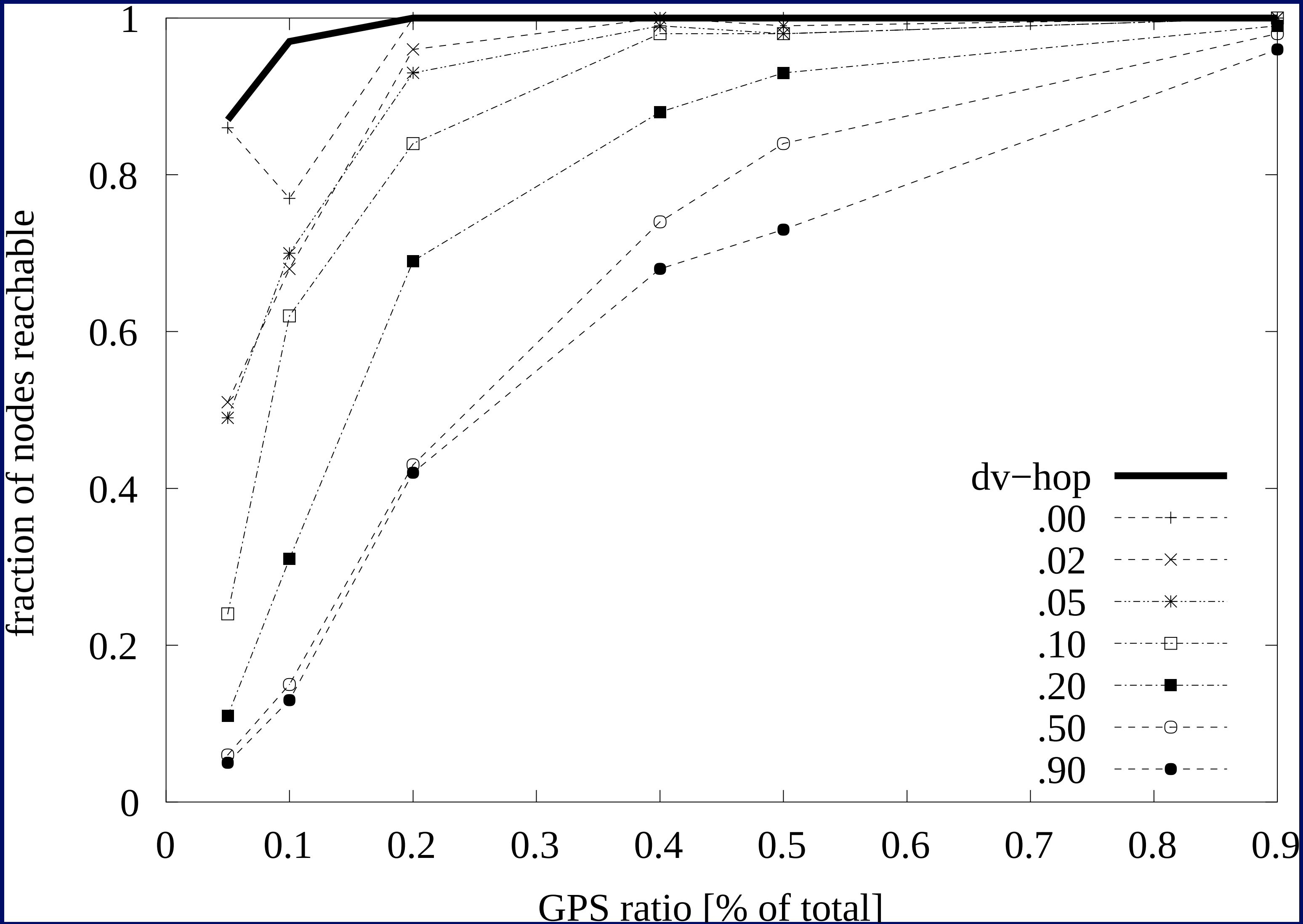


- *Euclidean* - error cumulates with distance
- *DV-based* - error cancels out over distance

reachability - *DV-distance*

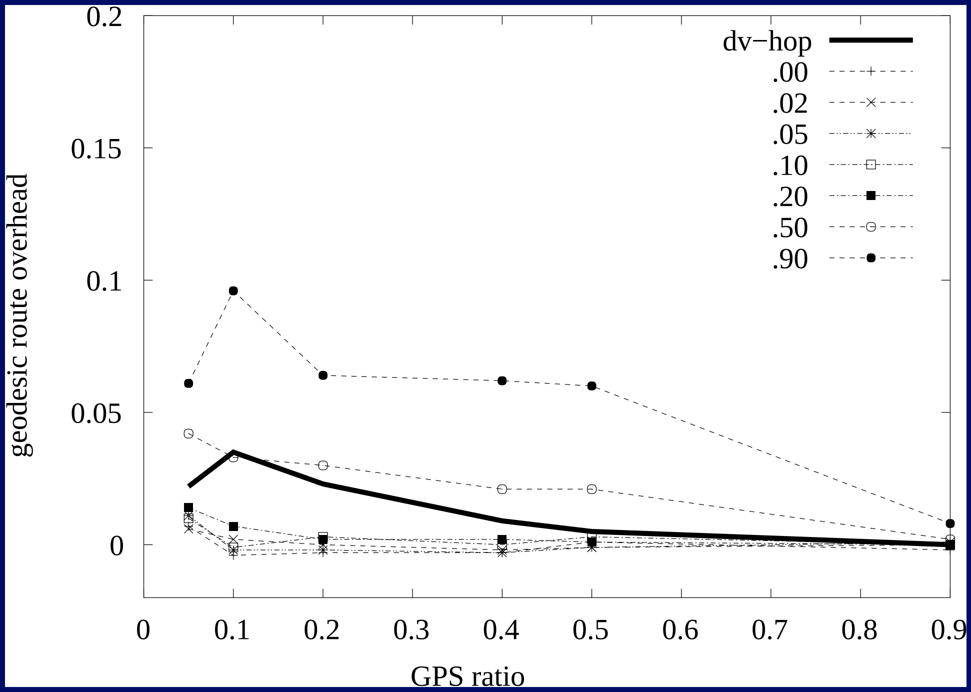


reachability - *Euclidean*

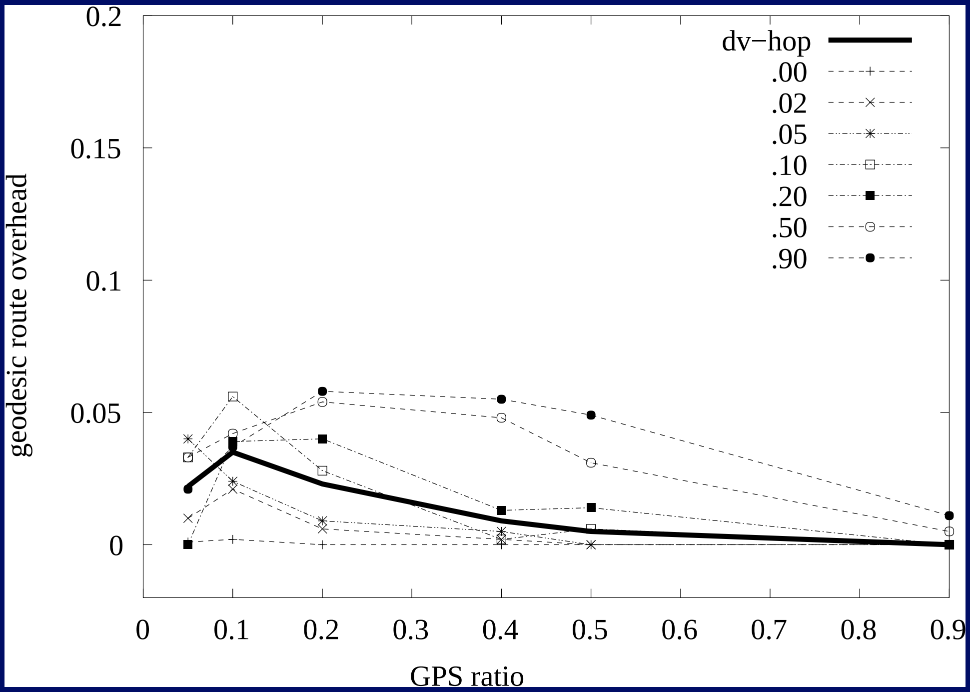


geodesic routing - overhead

DV-distance

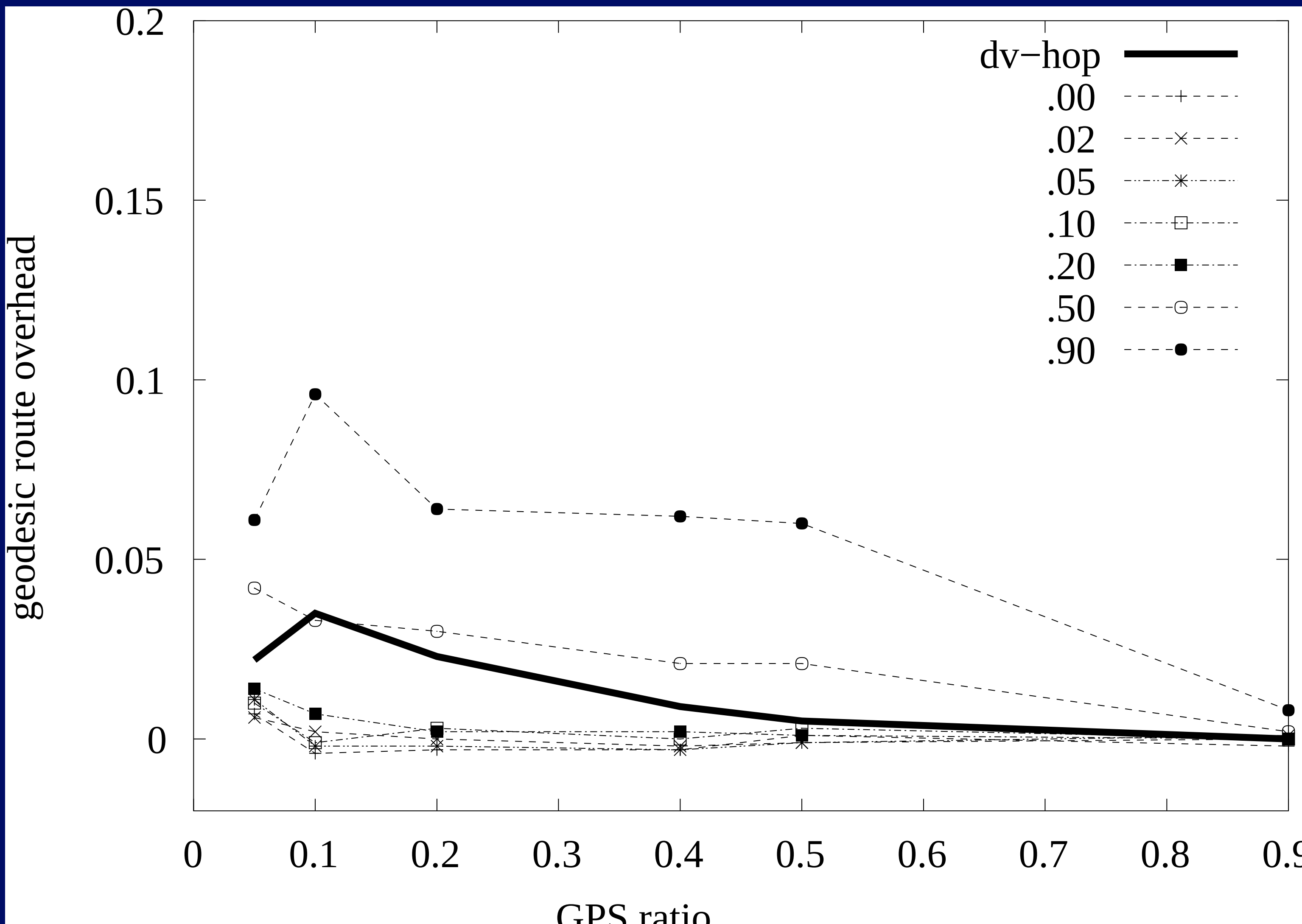


Euclidean

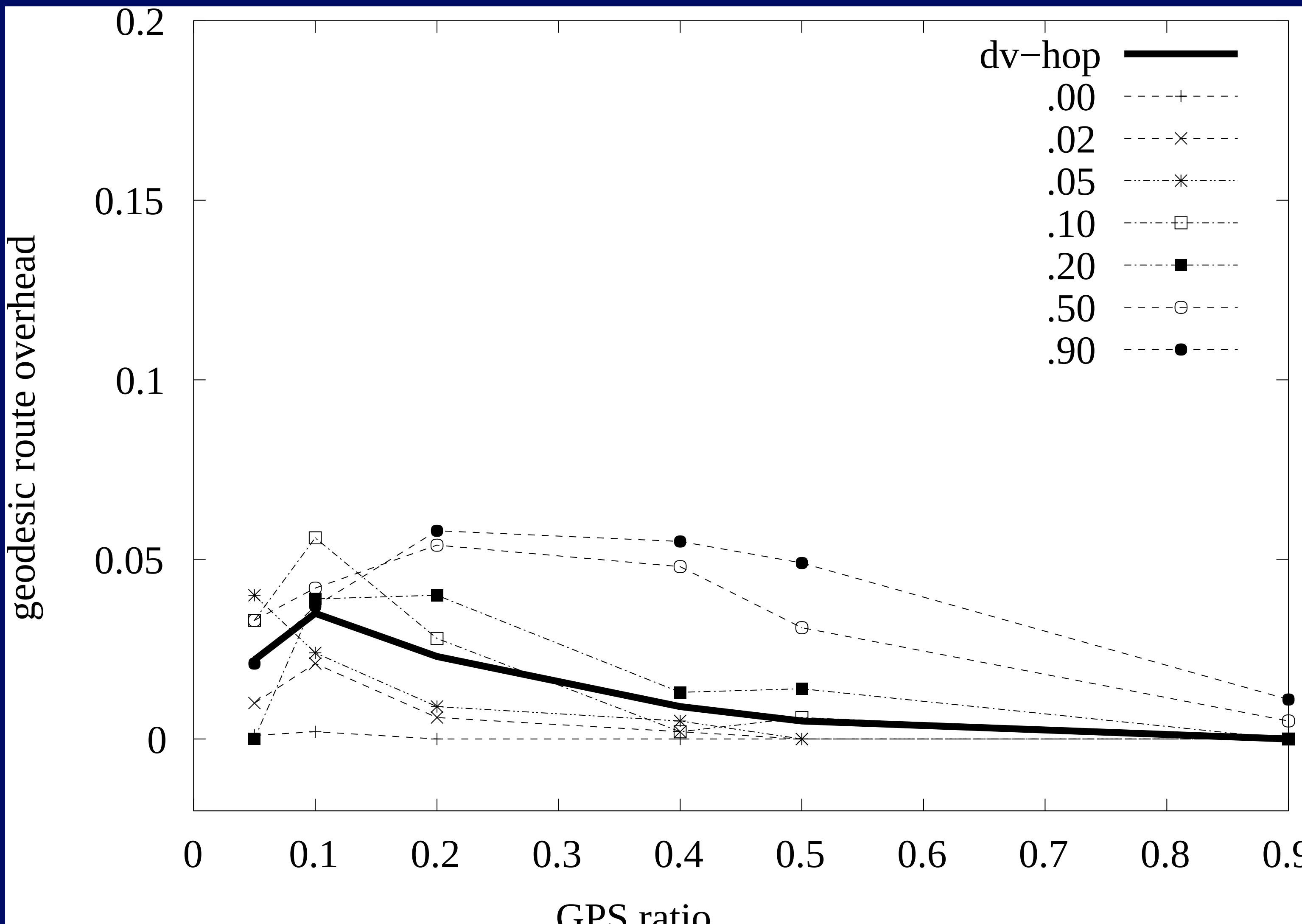


- even low overhead makes a difference in the long run

geodesic routing overhead - *DV-distance*



geodesic routing overhead - *Euclidean*



simulation summary

- all methods provide
  - usable locations for geodesic routing
  - location error with accuracy of 5%-50% of the radio range
  - better accuracy with more landmarks

DV-hop	*DV-distance*	*Euclidean*
isotropic ×	isotropic ×	nonisotropic √
high diameter √	high diameter √	low diameter ×
low GPS ratio √	low GPS ratio √	medium GPS ratio ×
immune to error, coarse √	error cancels out √	error builds up ×
√	more signaling due to measurement errors ×	more signaling for better coverage ×
2 floodings ×	2 floodings ×	1 flooding √
high variance ×	high variance ×	predictable perf. √

future work

- **node mobility**
  - a moving node needs to
    - get estimates from its new(static) neighbors
    - apply triangulation
  - a moving landmark
    - is a new landmark
    - one flying landmark could be enough for the entire network
  - mobile nodes are supported by static nodes
- **use AoA instead of signal strength**
  - having three angles to three known points → position

conclusions

- $APS = DV + GPS$
  - distributed
  - no infrastructure
  - recomputation only for moving nodes
- three propagation methods: *DV-hop, DV-distance, Euclidean*
  - there is a tradeoff between accuracy and signaling
  - there is a tradeoff between coverage and signaling
  - measurement error may affect signaling (*DV-distance*)
  - each is appropriate for different topologies and precision requirements