On Virtual Coordinate Based Routing and Performance of Random Routing in WSNs

Committee:
Prof. Anura P. Jayasumana
Prof. Ali Pezeshki
Prof. Indrakshi Ray

Dulanjalie Dhanapala

Thesis Defense
07.10.2009

Sponsors: NSF Grant CNS-0720889

Introduction

- Current routing protocols for WSNs
  - Address based
    - Structural or location based routing schemes
  - Content based
    - Random routing Flooding

Contribution

- **Virtual Coordinate Based Routing in WSNs**
  - Properties of VCS
  - Novel routing protocol- *Convex Subspace Routing*
  - Performance evaluation of CSR
- **Performance of Random Routing in Grid Based WSNs**
  - Analytical model
  - Model verification and applications

Introduction

- Routing Protocols
  - Address based
    - Hierarchical addressing
    - Physical coordinates
    - Virtual coordinates
  - Content based

- No geographical information
- GPS not feasible always
  - Position/location of the sensors
  - Energy constraints
- Routing → Insensitive to physical voids

Virtual Coordinate Systems - VCS

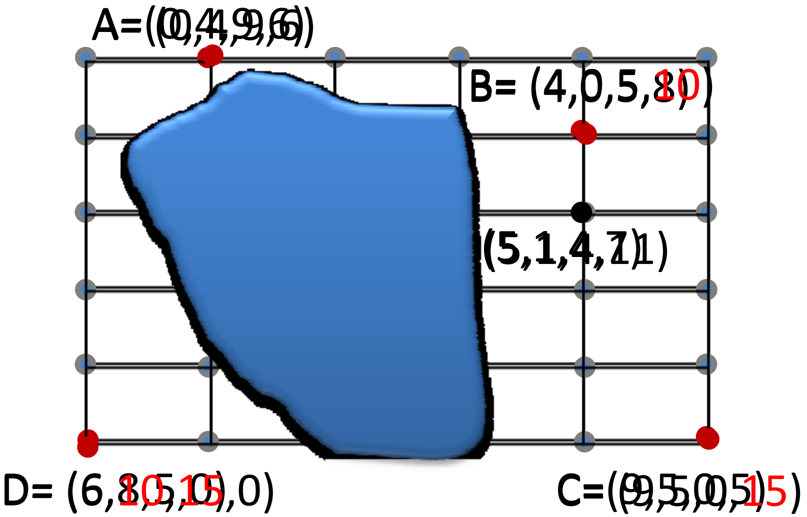
- Ordinate:

Relative position

in terms of # hops

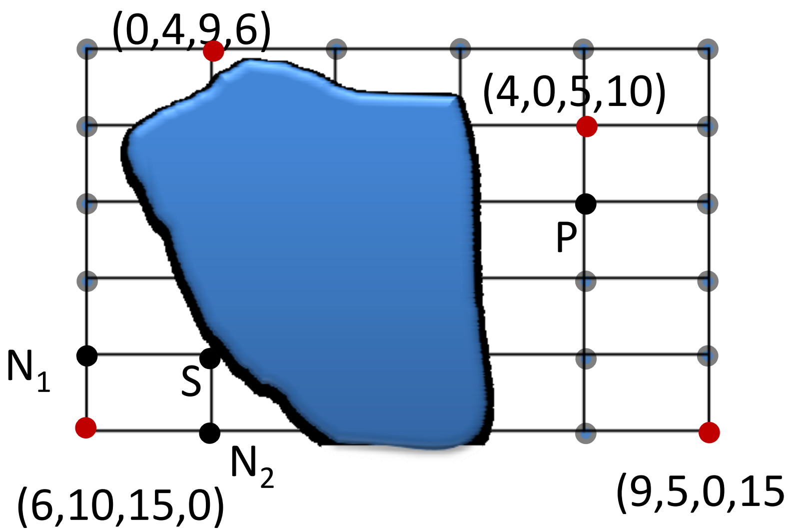
wrt an anchor node

  – Ex: A,B,C,D anchors



- Do not rely on geographic information
- Simple and scalable
- Routing is not sensitive to physical voids

Example



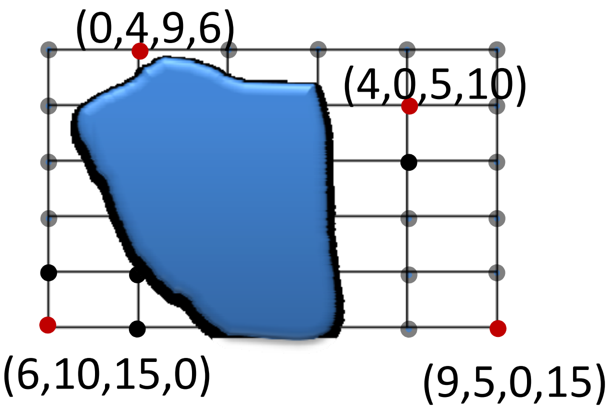
- P = (5,1,4,11)
- S = (6,10,15,2)
- N_1 = (5,9,14,1)
- N_2 = (7,11,16,1)
- n[S ,P] = 16.85
- n[N_1 ,P]= 16.25
- n[N_2 ,P]= 18.14

Forward to N_1

If geographical routing?

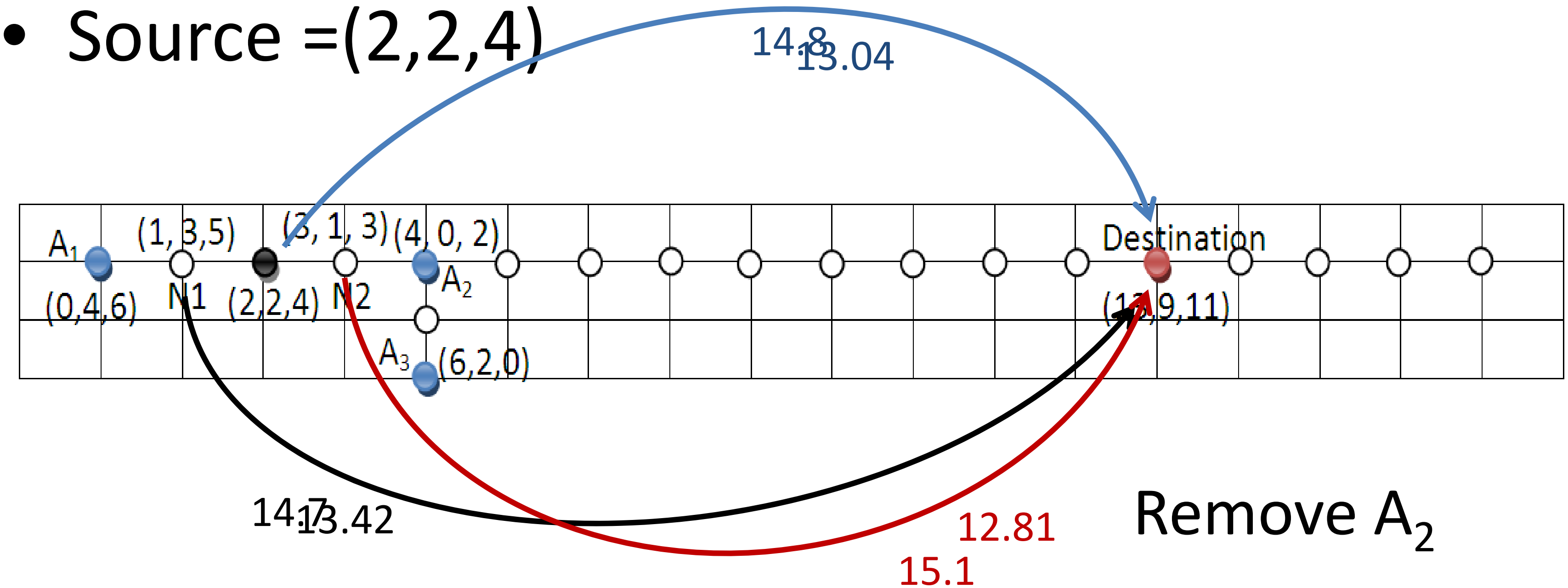
Issues in VCS

- Issue 1: Optimal number of anchors required is unknown
  - Under deployment of anchors
    - Identical coordinates
  - Over deployment of anchors
    - Inefficient
    - Redundant anchors → Redundant information
    - Degrade Greedy ratio (portion of paths that can be routed using GF only)



Example: Over Deployment of Anchors

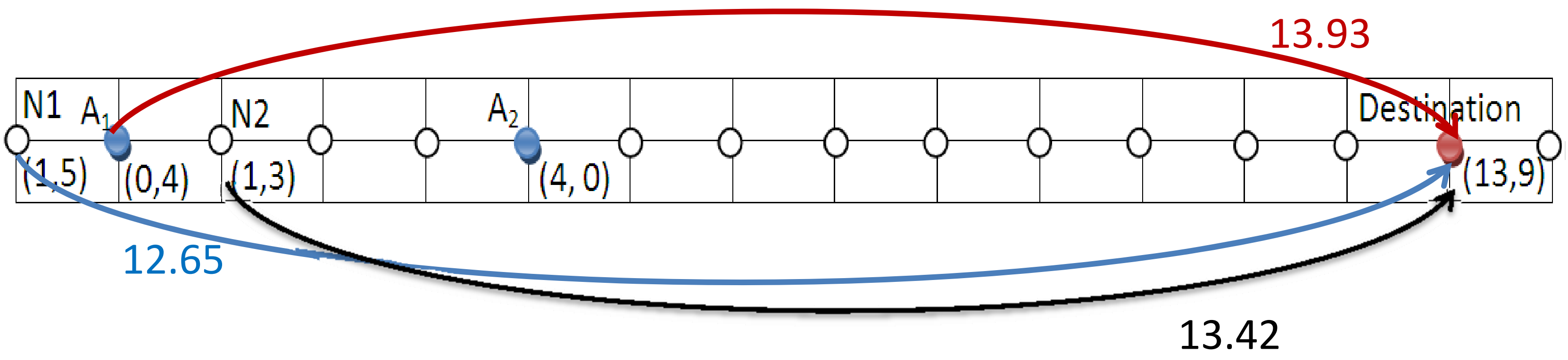
- Three anchors A_1 ,A_2 and A_3
- Source =(2,2,4)



- Redundant anchors give improper weight in some directions

Issue 2: Improper Anchor Placement

- Degrade Greedy ratio



- Also increases the identical coordinates

Properties of VCS

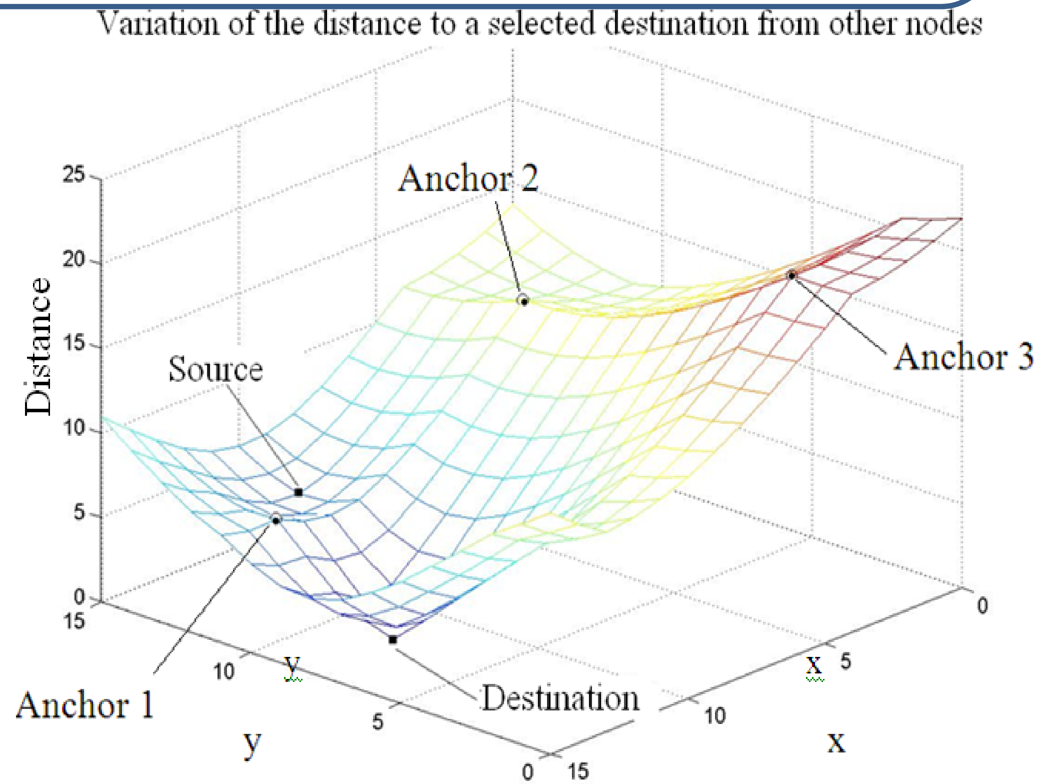
Property 1: In a virtual coordinate system, two anchors cannot have identical coordinates. Also a node and an anchor coordinate cannot have identical coordinates

- i^{th} anchor's , i^{th} ordinate is zero
  - Ordinates are always positive
  - Orthogonal coordinate system

Properties of VCS (Cntd.)

Property 2: Internal anchors are local maximizers in distance function corresponding to its own coordinate

Variation of distance to a selected destination from all the other nodes in a 15x15 grid with three anchors in the grid



Illustrate Property 2

- Network with single anchor A
- Destination, N_d=(n[A, N_d])
- Any other node's coordinate = n
- Distance function from any node N_i to node N_d

$$n[N_i, N_d] = \sqrt{(n - n[A, N_d])^2}$$

Illustrate Property 2 (Cntd.)

$$\underset{n}{argmax}\left(\sqrt{(n - n[A, N_d])^2}\right) = 0$$

- Zero is anchor coordinate
- Two anchors A_1 and A_2 ;

$$(n[N_i, N_d])^2 = (n[N_i, A_1] - n[N_d, A_1])^2 + (n[N_i, A_2] - n[N_d, A_2])^2$$

- 1st term alone creates a maximum at anchor A_1 and 2nd term at anchor A_2

Internal anchors may cause local maxima and identical coordinates

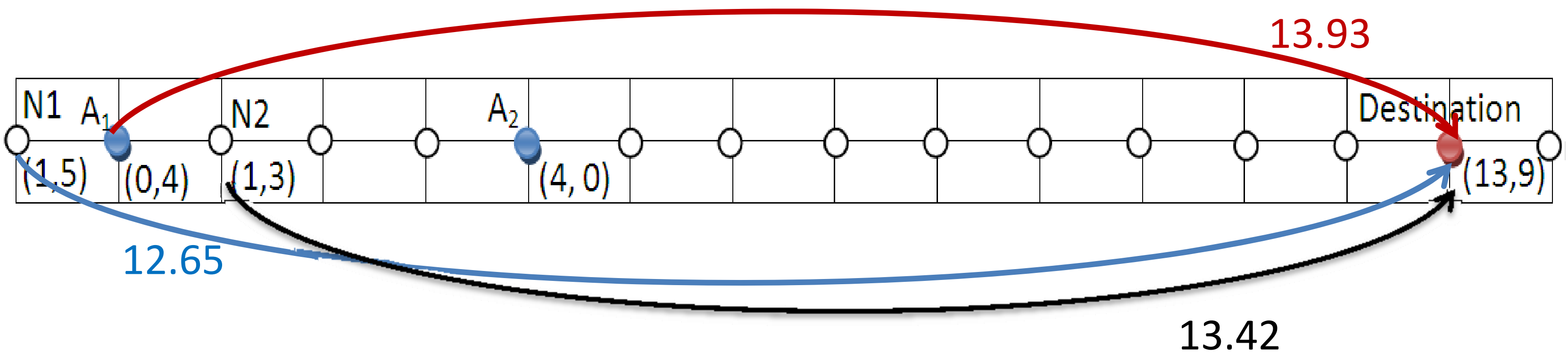
Proper Anchor Placements in 1-D Network

Lemma 1:

- One anchor placed at the corner of a 1-D network
  - provides unique coordinates for different nodes
  - allows for routing without local maxima achieving 100% greedy ratio
- If two anchors are placed in the middle
  - they are able to provide unique coordinates
  - Yet they introduce local maxima and minima in distance

Proof

- First part is obvious
- Ex: if two anchors are placed in the middle



Proper Anchor Placements 2-D Full Grid

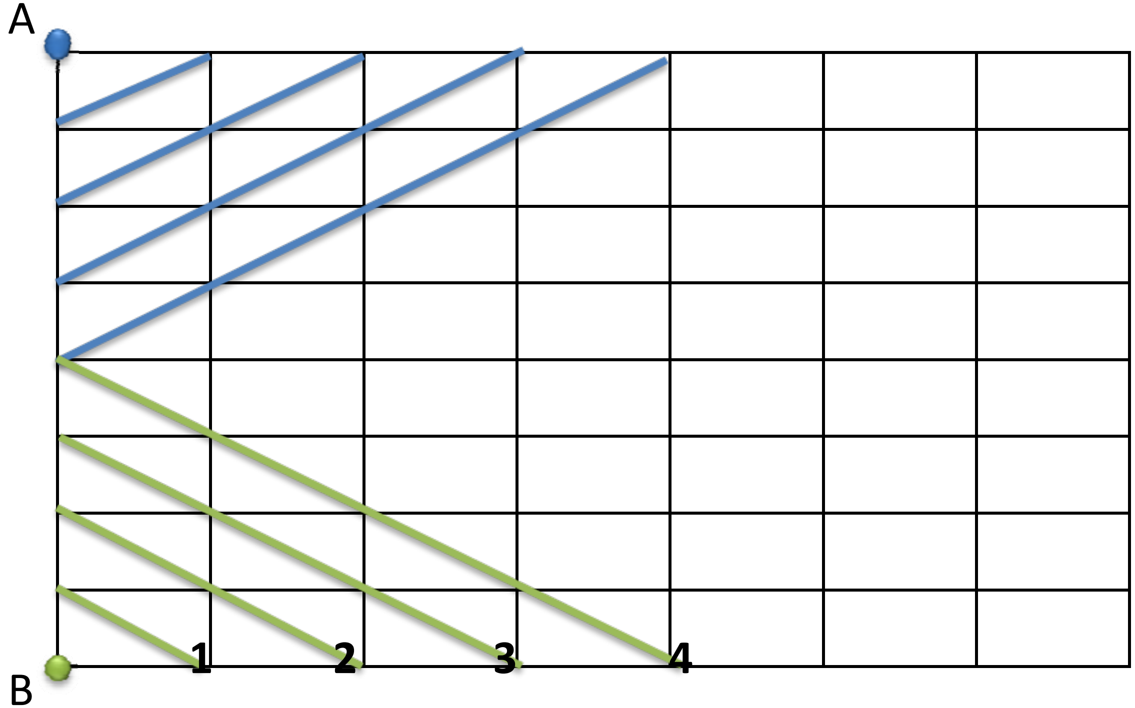
Lemma 2:

- For a rectangular full grid, two nodes placed at adjacent corners are sufficient to uniquely name all the nodes
- Furthermore, such a coordinate system does not introduce local maxima or minima in distance space, resulting in a greedy ratio of 100%

Proof - Part I

A and B are anchors

- Nodes are at all the cross points
- Blue and green lines: level sets with respect to A and B



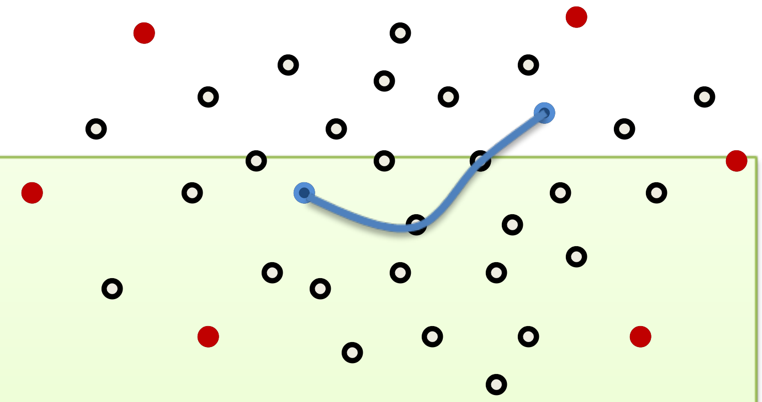
Proof – Part 2

- 100% greedy ratio
- Distance function is parabolic with minimum at destination

$$(n[N_s, N_d])^2 = (x + y - x_d - y_d)^2 + (x - y + N - (x_d - y_d + N))^2$$

$$= 2(x - x_d)^2 + 2(y - y_d)^2$$

Upper and Lower Bounds for Path Lengths



Lemma 3:

- M anchors
- Source ≡ (n[N_s ,A_1], n[N_s,A_2],…)
- Destination ≡ (n[N_d,A_1], n[N_d,A_2],…)
- Shortest hop distance between the two nodes in hops, Min(n[N_s, N_d]) is bounded by:

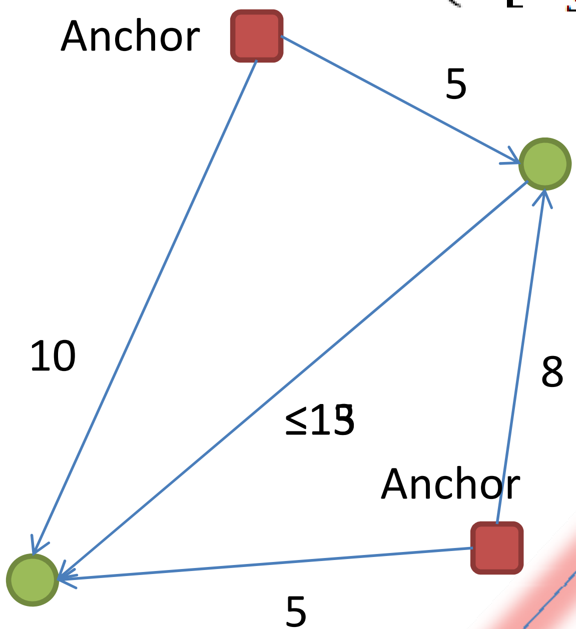
$$\mathrm{Max}\,(|\mathrm{n}[N_s, \mathrm{A_i}], -\mathrm{n}[N_d, \mathrm{A_i}]|) \leq \mathrm{Min}(\mathrm{n}[N_s, N_d]) \leq \mathrm{Min}(\mathrm{n}[N_s, \mathrm{A_i}] + \mathrm{n}[N_d, \mathrm{A_i}]);$$

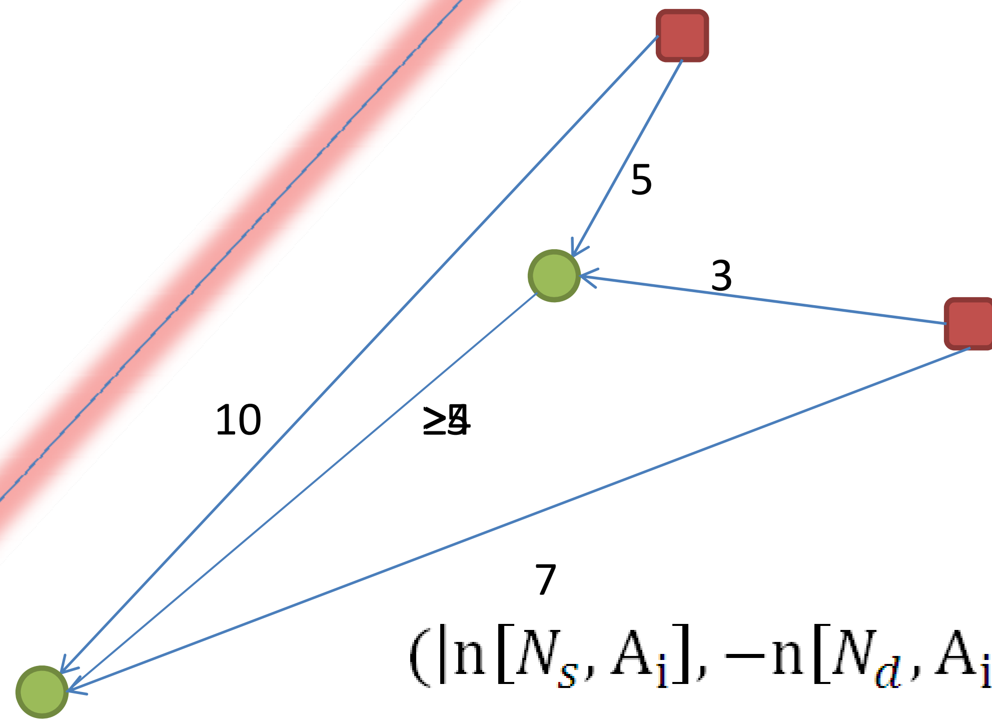
Proof

$$\text{Max}\,(|\text{n}[N_s, \text{A}_\text{i}], -\text{n}[N_d, \text{A}_\text{i}]|) \leq \text{Min}(\text{n}[N_s, N_d]) \leq \text{Min}(\text{n}[N_s, \text{A}_\text{i}] + \text{n}[N_d, \text{A}_\text{i}]);$$

$$\text{n}[N_s, \text{A}_\text{i}] + \text{n}[N_d, \text{A}_\text{i}]$$

$$\text{Min}(\text{n}[N_s, \text{A}_\text{i}] + \text{n}[N_d, \text{A}_\text{i}])$$



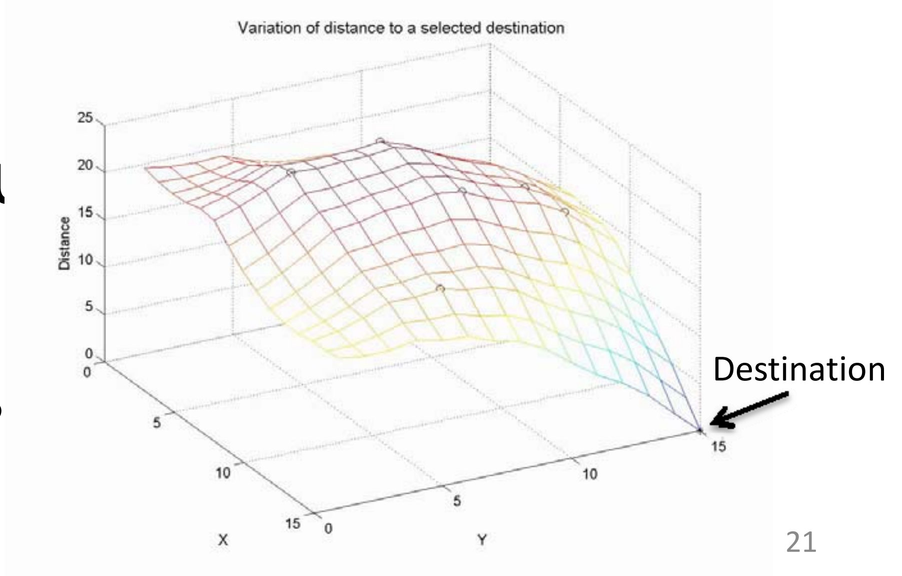


$$(|\text{n}[N_s, \text{A}_\text{i}], -\text{n}[N_d, \text{A}_\text{i}]|)$$

$$\text{Max}\,(|\text{n}[N_s, \text{A}_\text{i}], -\text{n}[N_d, \text{A}_\text{i}]|)$$

Improvement in Routability: Convex Subspace Routing (CSR)

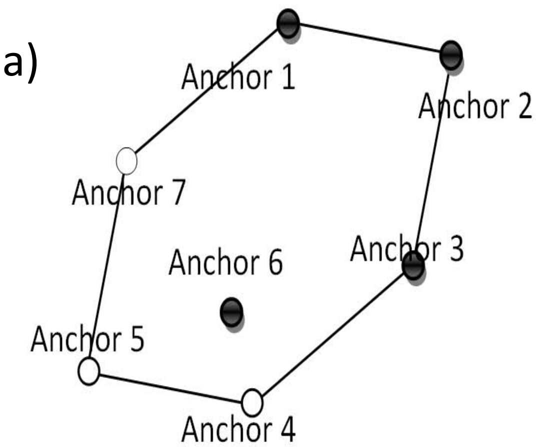
- No need of back tracking if distance surface is convex
- Select subset of anchors
  - convex distance function from source to destination
- *M* anchors→ select su
  - *r*, vertices of a convex
  - Current node and des convex set



Example: Convex Subspace

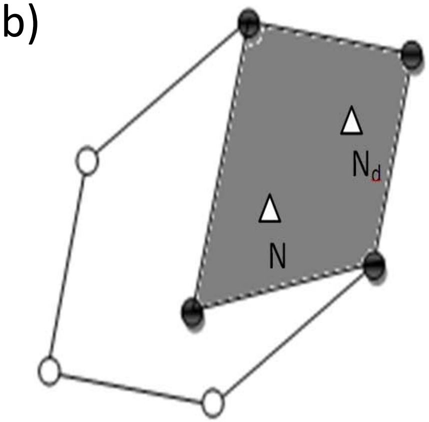
- *M=7, r = 4*

a)



a)convex polygon created by 7 anchors

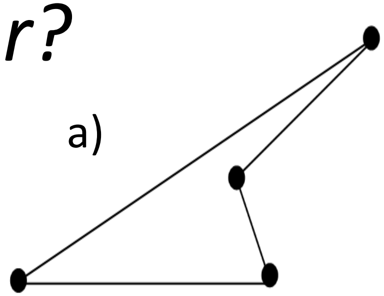
b)



b) convex polygon created by subset (4) of anchors

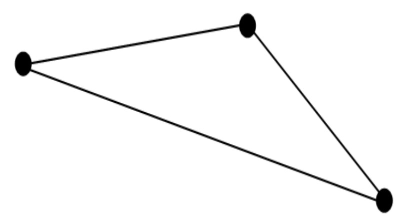
- What is the value of *r?*
  - Three. Why?

a)



a) Non convex boundary created by 4 anchors

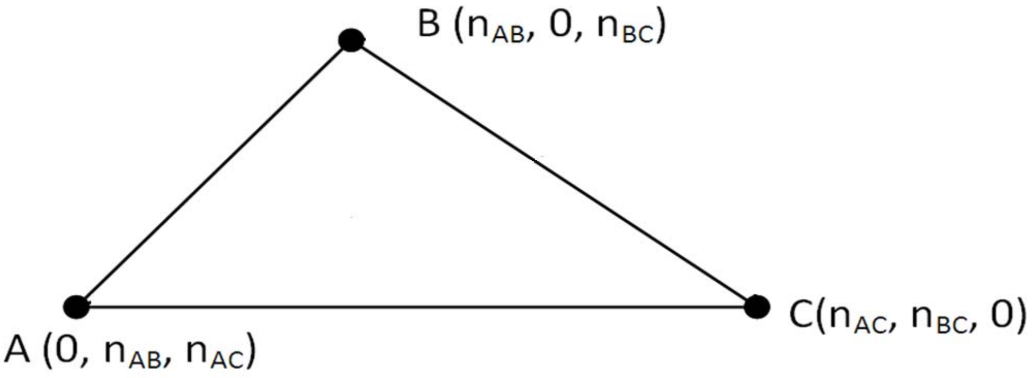
b)



b) Triangle is always convex shape

Identifying Three Anchors that Enclose a Node

- In **virtual space**, 3 anchors will give a **triangle**



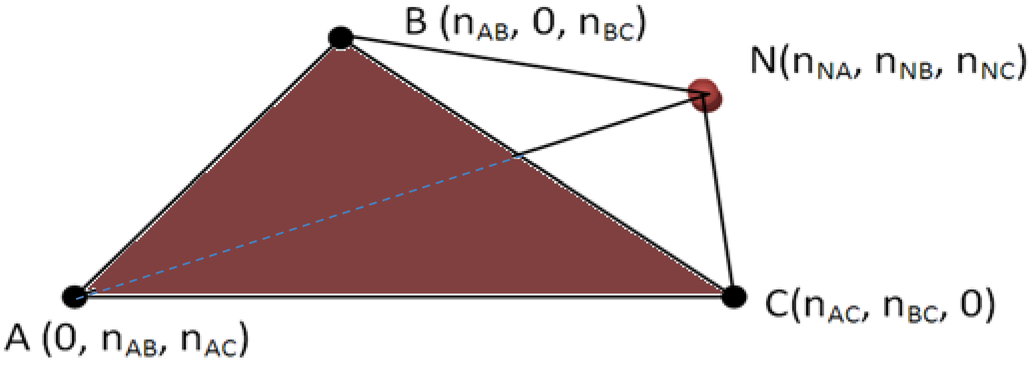
- Area of a triangle of perimeter 2*S*;

$$\sqrt{S(S - n_{AB})(S - n_{BC})(S - n_{AC})}$$

$$S = \frac{1}{2}(n_{AB} + n_{BC} + n_{AC})$$

Identifying Three Anchors that Enclose a Node (Cntd.)

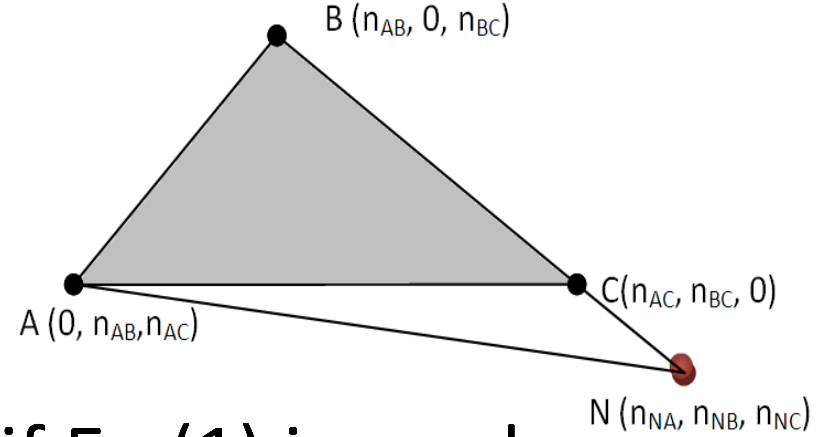
- Any node N



- If N is inside ABC, then

$$Max[(\Delta NAB + \Delta NAC), (\Delta NAB + \Delta NBC), (\Delta NBC + \Delta NAC)] \leq (\Delta ABC) \quad (1)$$

Identifying Three Anchors that Enclose a Node (Cntd.)

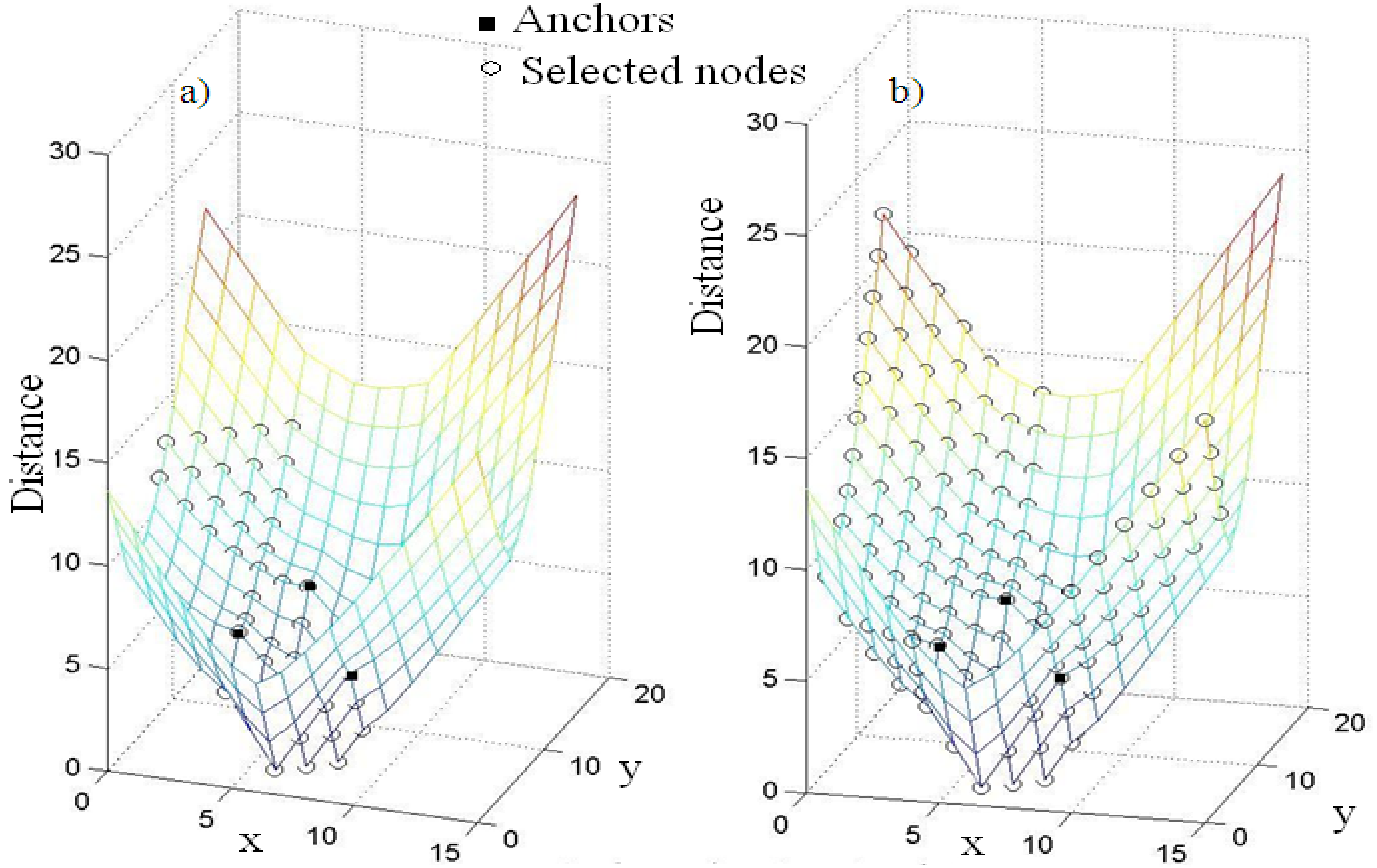


- *N* will not be captured if Eq (1) is used
- But *N* is also in the routable set

$$Min[(\Delta NAB + \Delta NAC), (\Delta NAB + \Delta NBC), (\Delta NBC + \Delta NAC)] \leq (\Delta ABC) \quad (2)$$

- Larger feasible set

Example



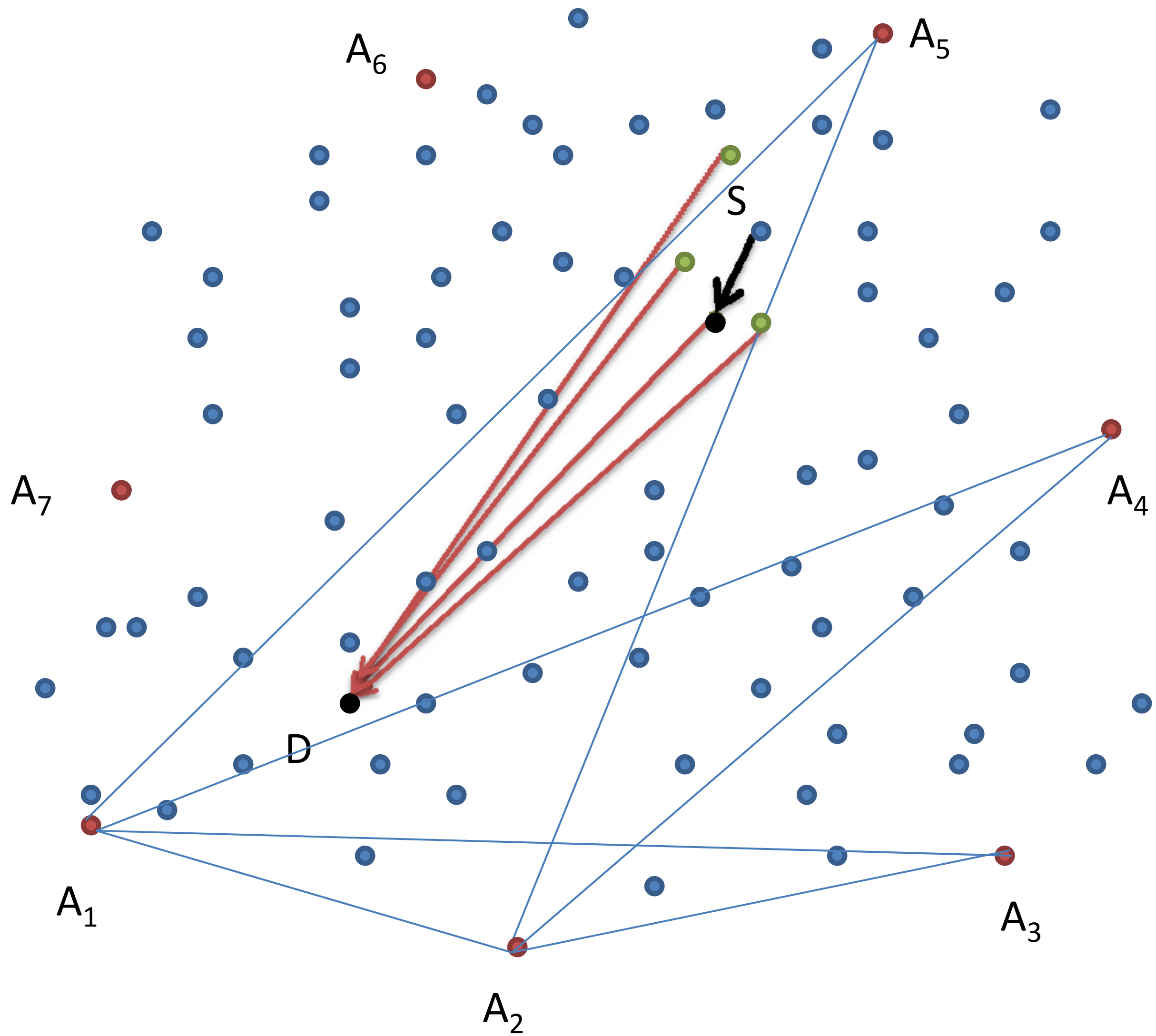
Distance function to a selected destination a) Max [(ΔNAB+ΔNAC), (ΔNAB+ ΔNBC), (ΔNBC+ΔNAC)]≤ (ΔABC) b) Min[(ΔNAB+ΔNAC), (ΔNAB+ΔNBC), (ΔNBC+ΔNAC)] ≤ (ΔABC)

Algorithm of CSR

if *(*N_d *is not* N_i*)*
    while*(*N_d *is not reached)*
        Find the 1st ***suitable*** *triplet of anchors that includes* N_i *and* N_d
        if *a triplet NOT found*
            Routing failed
        else
            Evaluate the distances from N_i *and its neighbors to* N_d *using only the coordinate with respect to selected triplet*
            if *min(distances (neighbors to* N_d*))==0*
                if *neighbor that has zero distance ==* N_d
                    Successfully routed
                else
                    Get another triplet for routing. If no triplet found then routing failed
                end
            elseif *min(distances(a neighbor to* N_d*))≤ distances (*N_i *to* N_d*)*
                N_i*= neighbor that has the minimum distance*
            else *%i.e. (distances (neighbors to* N_d*))> distances (*N_i *to* N_d*)*
                Get another triplet for routing. If no triplet found then routing fail
            end
        end
    end
end

Algorithm of CSR(Cntd.)

A_1, A_2, A_5

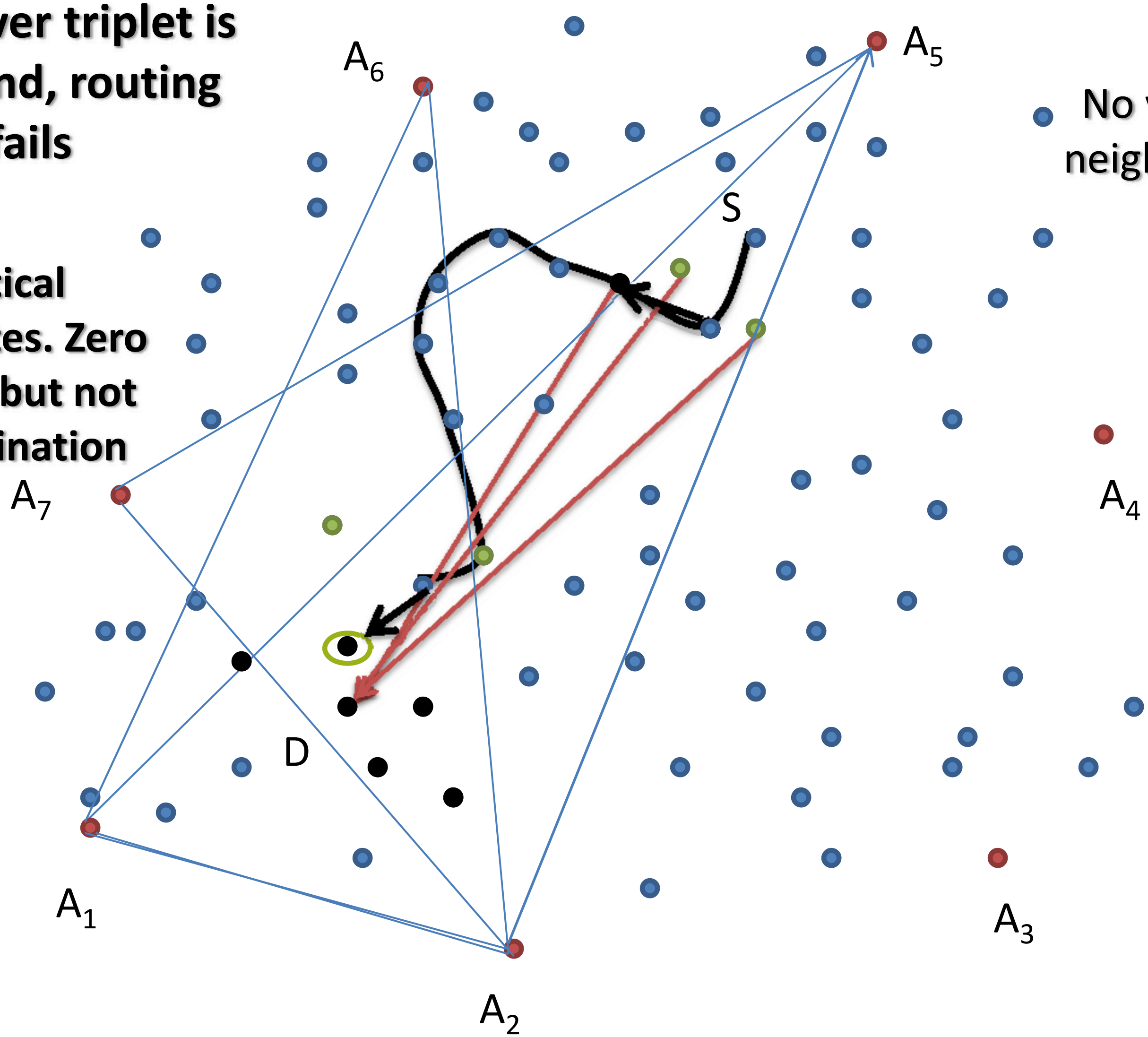


Algorithm of CSR(Cntd.)

When ever triplet is not found, routing fails

Identical coordinates. Zero distance but not the destination

No virtually closer neighbor → Change the triplet

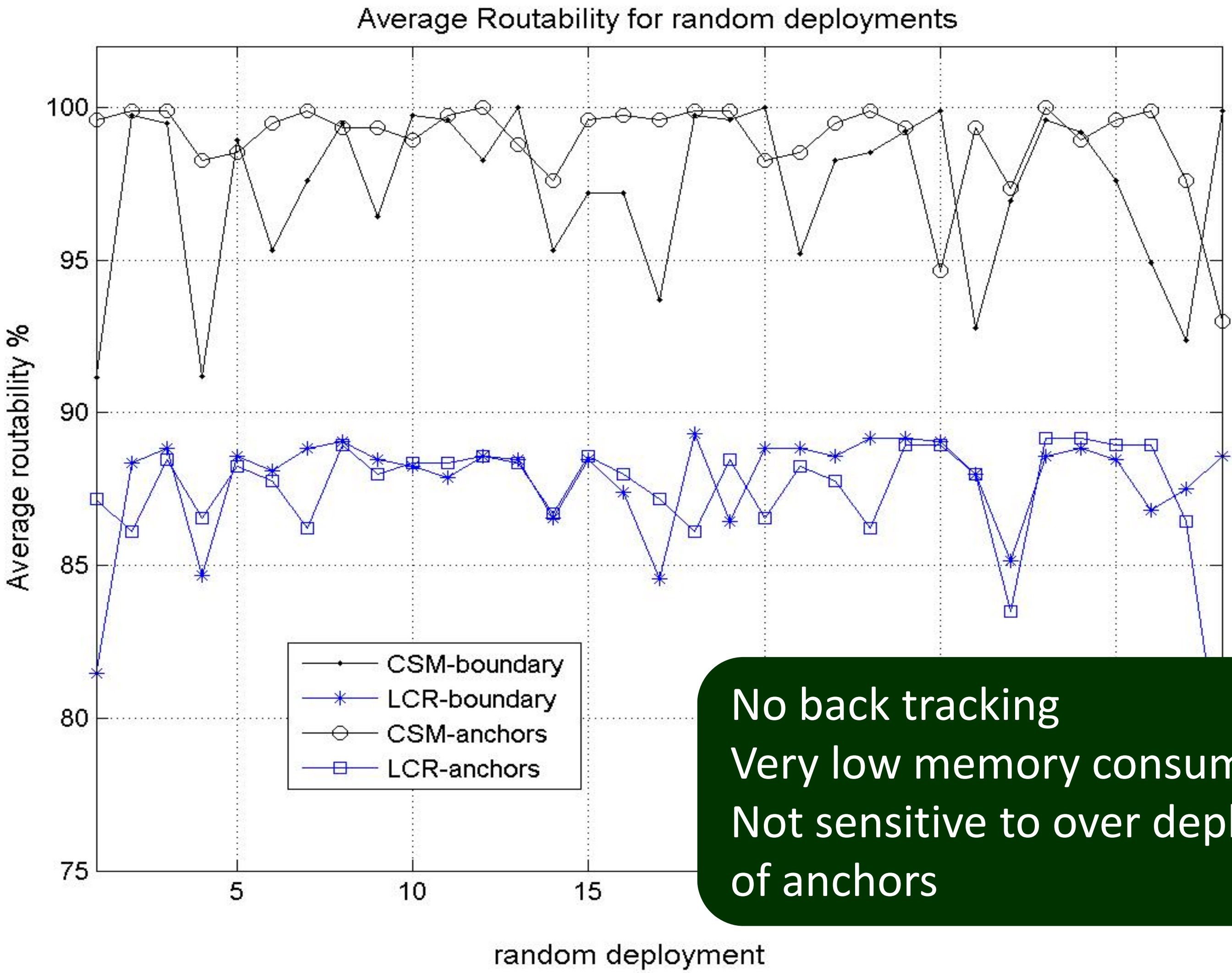


Simulation Results

- 30 x 30 grids with 100 missing nodes
  - Randomly placed
- 20 anchors
  - Randomly placed
    - On the boundary
    - Anywhere
    - Furthest apart property is not considered
- Compared with Logical Coordinate Routing (LCR) [1]

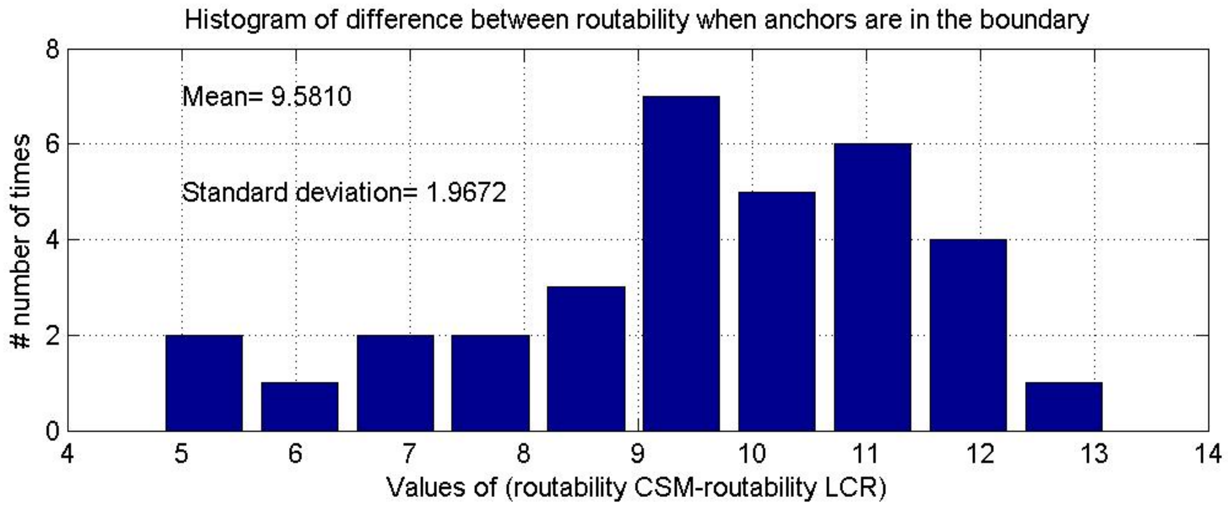
[1] Q. Cao and T Abdelzaher, "Scalable logical coordinates framework for routing in wireless sensor networks", ACM transactions on Sensor Networks, Vol. 2,No. 4,pp. 557-593, Nov 2006.

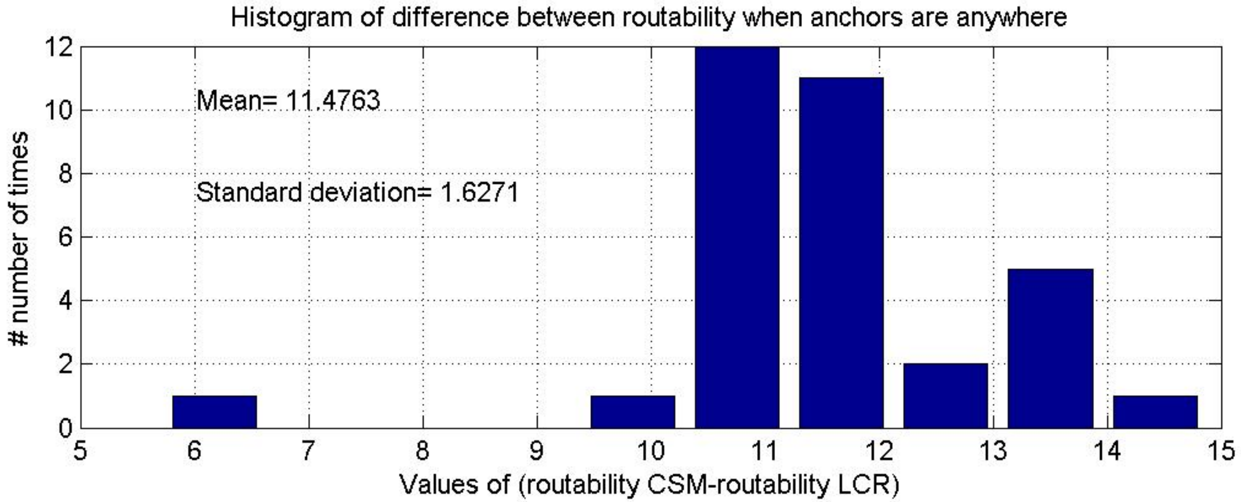
Performance Variation with Random Deployments



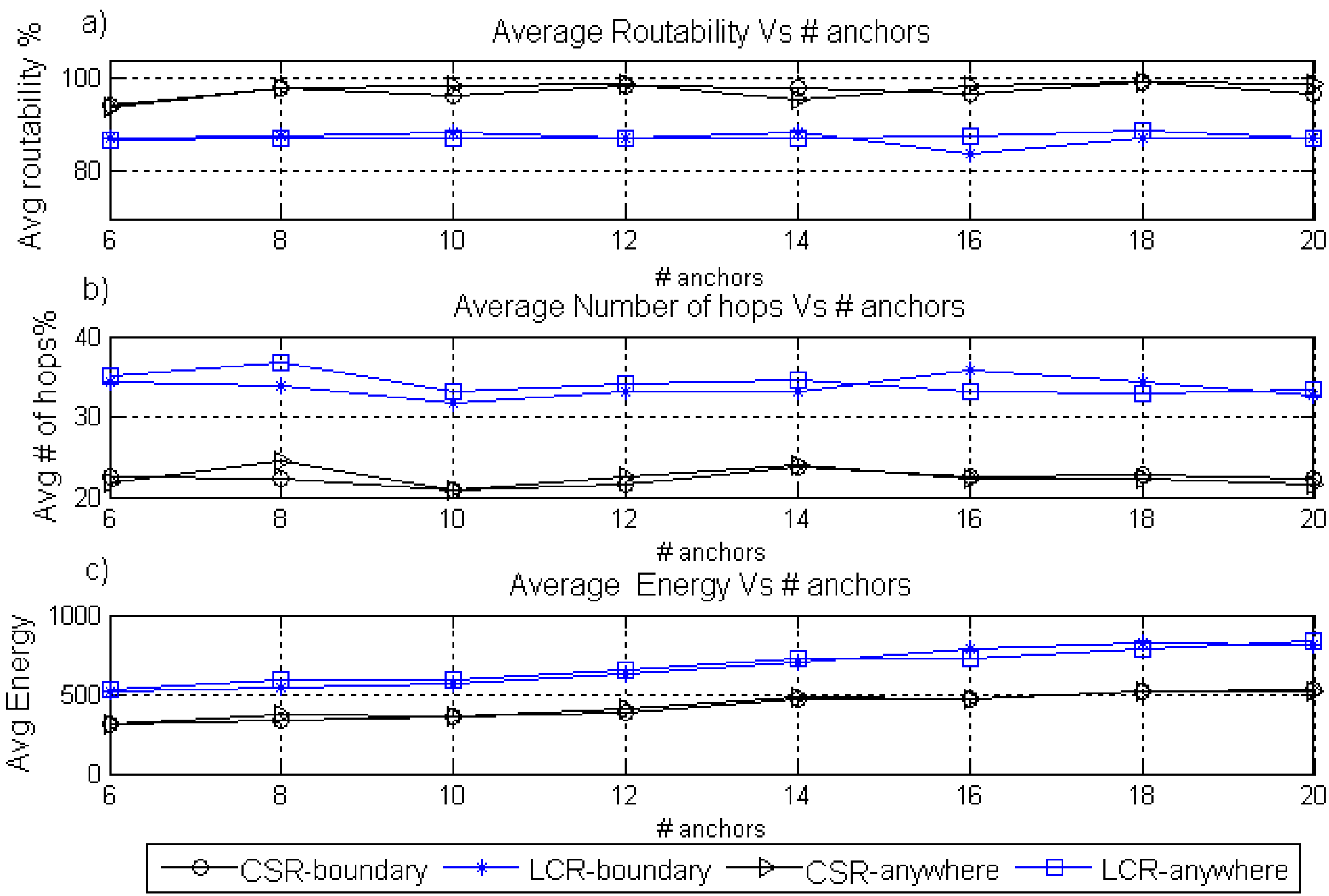
No back tracking
Very low memory consumption
Not sensitive to over deployment of anchors

Performance Variation with Random Deployments (Cntd.)

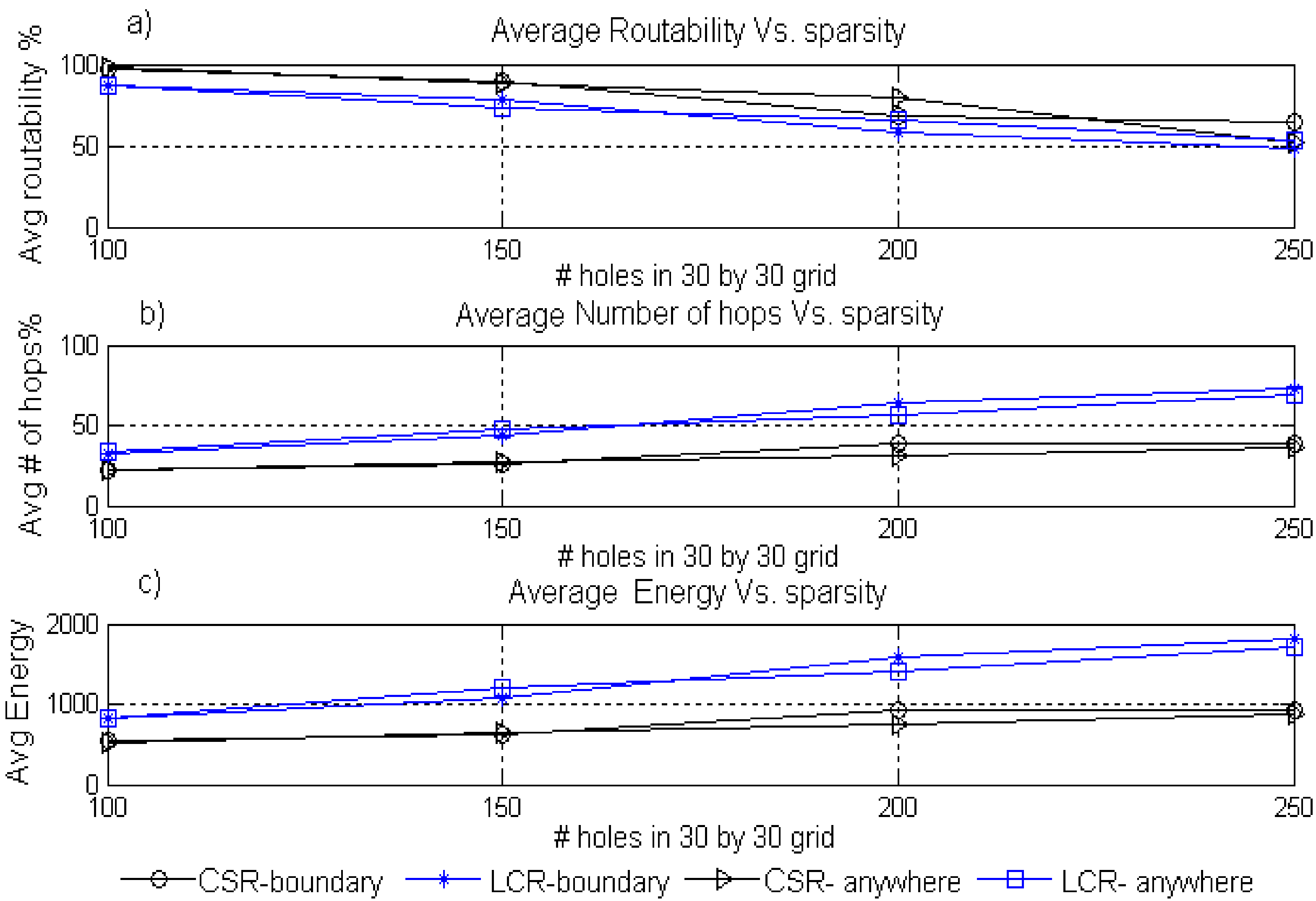




Performance Variation with Number of Anchors



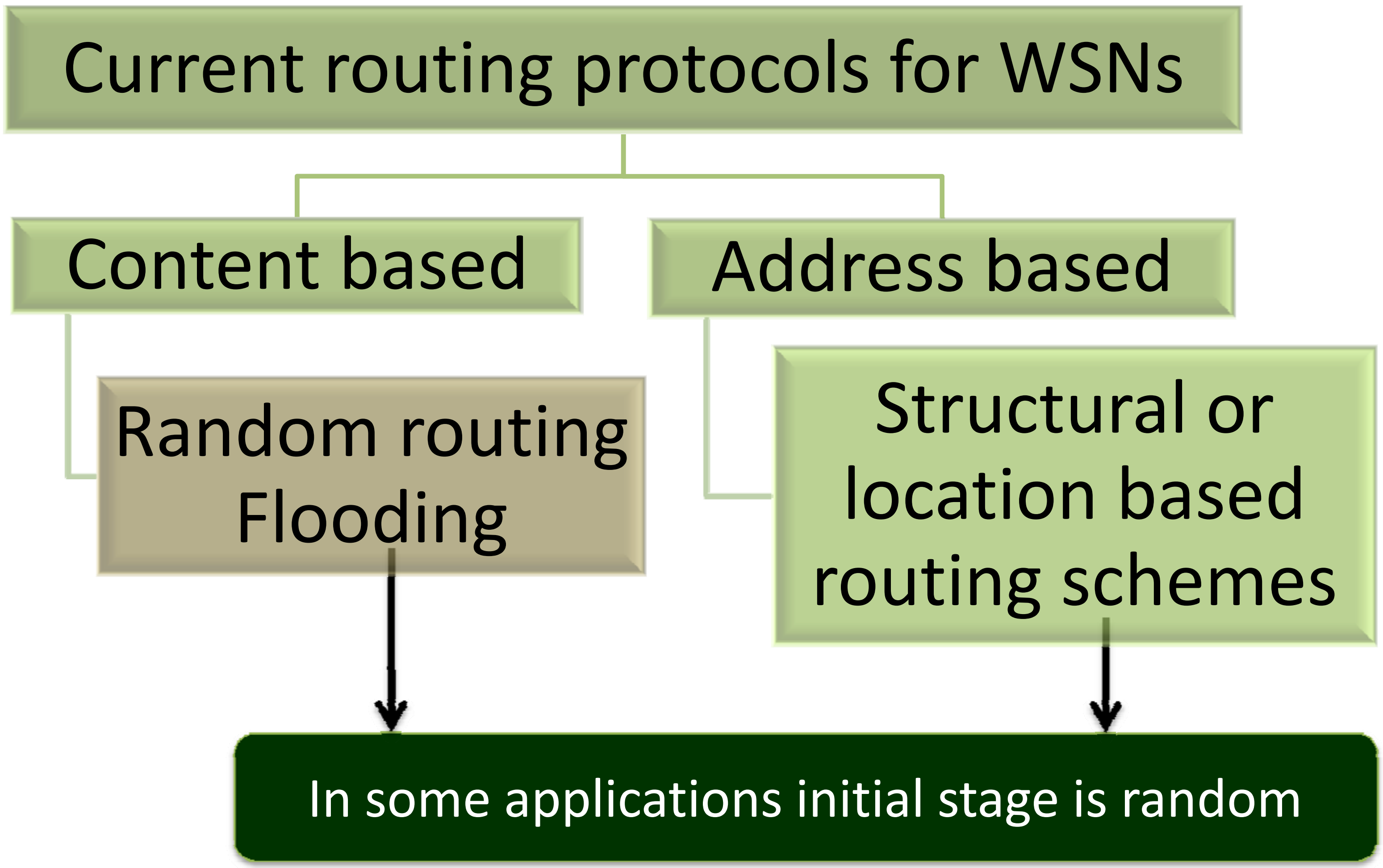
Performance Variation with Sparsity



Contribution

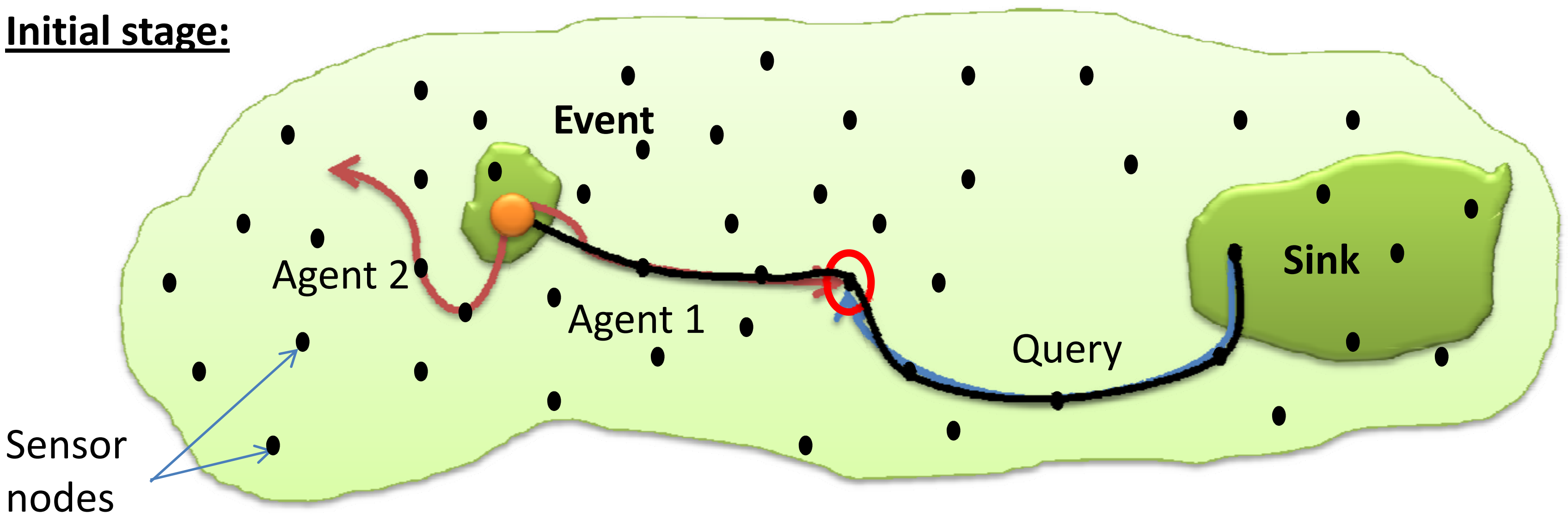
- **Virtual Coordinate Based Routing in WSNs**
  - Properties of VCS
  - Novel routing protocol- *Convex Subspace Routing*
  - Performance evaluation of CSR
- **Performance of Random Routing in Grid Based WSNs**
  - Analytical model
    - For 5 cases
  - Model verification and applications
    - For 3 applications

Introduction: Performance of Random Routing in Grid Based WSNs



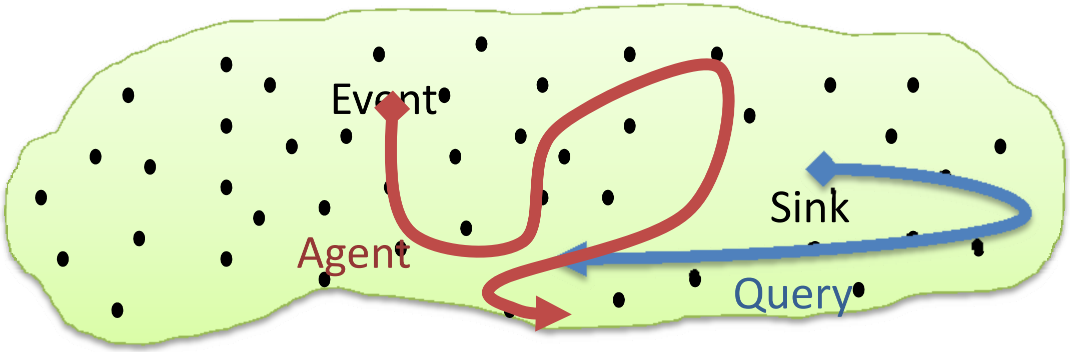
Introduction (Cntd.)

Initial stage:



Contribution

- Mathematical model to evaluate
  - **Exact** probability of a packet visiting a node within a given number of hops
  - Rendezvous probability of agent and query
  - Optimize the # of queries/agents required under different constraints

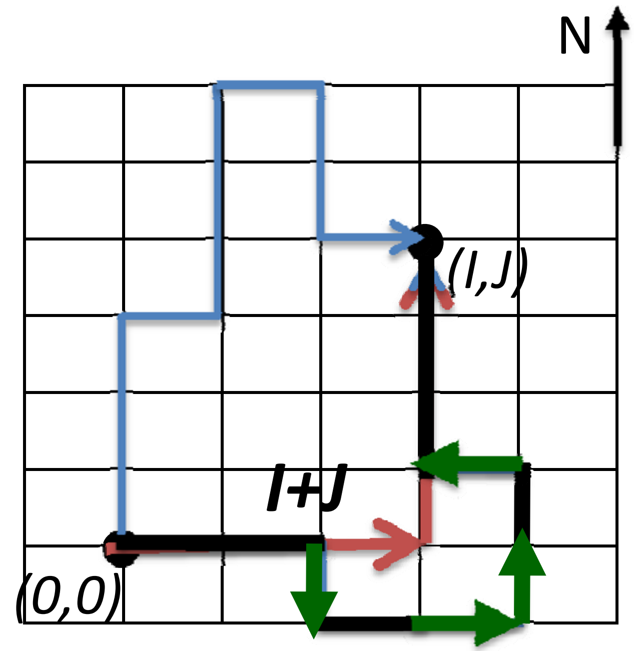


Analytical Model

- *Step 1: P_H (I,J) :* P[Packet reaching (*I,J*) in the *H*-th hop]
- *Step 2: Q_H(I,J)* : P[Packet visiting *(I,J)* within H-hops]
- *Step 3:* P[Agent meeting query anywhere for the first time within h_q-hops]

Step 1: Packet reaching (*I,J*) in the *H*-th hop

- Number of hops moved in
  - East(E)=*e*, West(W)=*w*,
  - North(N)=*n*, South(S)=*s*



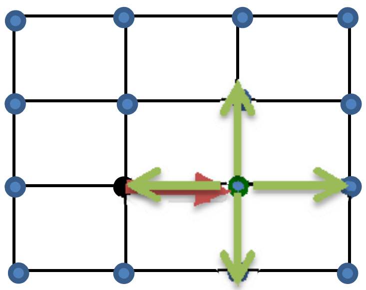
I+J + *H-(I+J)* = *H*

H							
$I+J$				$H-(I+J)$			
$e = I$	$w = 0$	$n = J$	$s = 0$	$e = i$	$w = i$	$n = j$	$s = j$

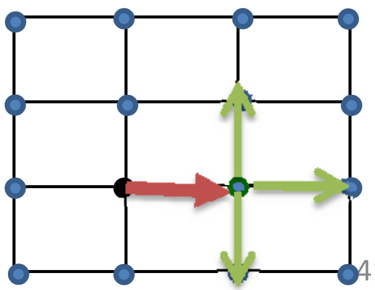
Step 1 (Cntd.)

Select the next node with:

- Case 1: Equal probability



- Case 2: Equal probability in a lossy network
- Case 3: Equal probability in lossless networks with rectangular boundaries
- Case 4: Unequal probabilities
- Case 5: Self avoiding forwarding



Case 1: Select the Next Node With Equal Probability

- $P_H(I,J)$: Packet reaching (I,J) in the H-th hop

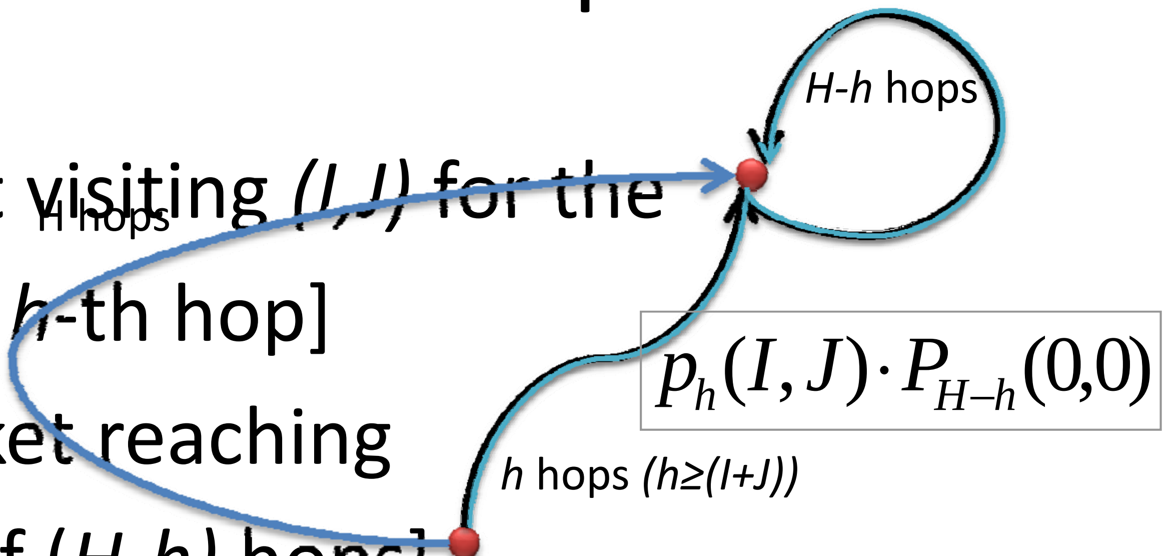
$$P_H(I,J) = \sum_{i=0}^{(H-K)/2} \frac{H!}{(I+i)!\,i!\,(J+j)!\,j!}\left(\frac{1}{4}\right)^H ; i+j=(H-(I+J))/2$$

- Using Vandermonde's Convolution

$$P_H(I,J) = \frac{H!\,{}^{H}C_M}{(M-I)!(M-J)!}\left(\frac{1}{4}\right)^H ; M=\frac{H+I+J}{2}$$

Probability of a Packet Visiting *(I, J)* in the *H*-th Hop – Another Representation

- $p_h(I,J)$: P[Packet visiting *(I,J)* for the first time in the *h*-th hop]
- $P_{H-h}(0,0)$: P[Packet reaching *(0,0)* at the end of *(H-h)* hops]
- $P_H(I,J)$: P[Packet reaching *(I,J)* in the *H*-th hop]

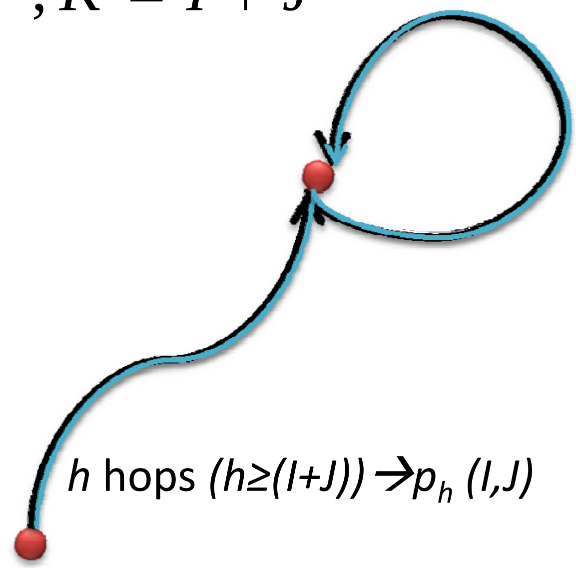


$$P_H(I,J) = \sum_{i=0}^{H} p_{H-i}(I,J)\cdot P_i(0,0)$$

Step 2: Packet visiting *(I, J)* within *H*-hops

- $Q_H(I,J)$: P[Packet visiting *(I,J)* within H-hops]

$$Q_H(I,J) = \sum_{h=K}^{H} p_h(I,J) \quad ; K = I + J$$



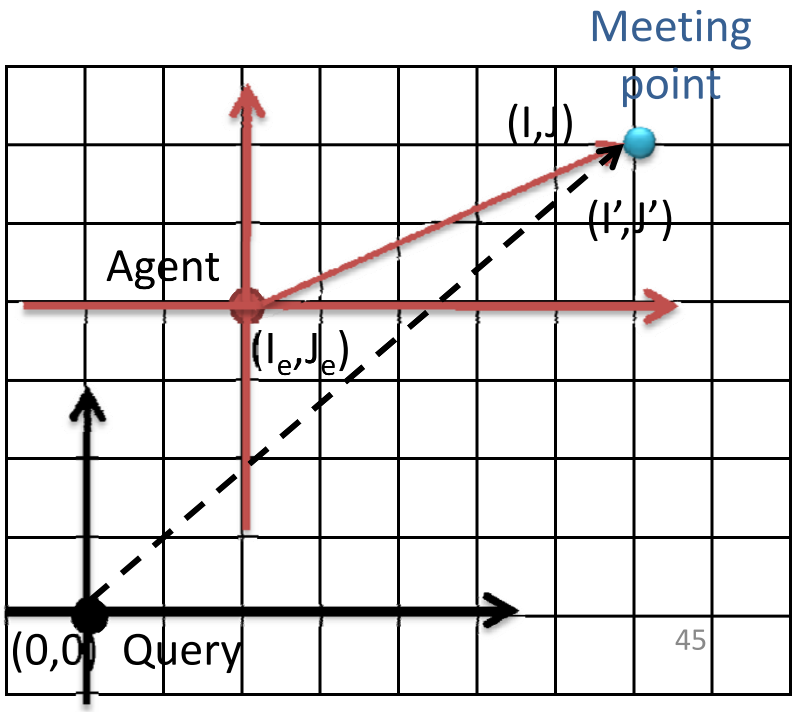
Step 3: Rendezvous Probability of Agent and Query

- P [Agent NOT meeting query anywhere within h_q-hops]

$$M_{H_e,h_q} = \prod_{\forall (I+J)\le h_q} \left(1 - Q_{H_e}(I,J) Q_{h_q}(I',J')\right)$$

- P [Agent meeting query anywhere for the first time within h_q-hops]

$$R_{H_e,h_q} = 1 - M_{H_e,h_q}$$



Step 3 (Cntd.)

- $Q_h^{(N)}(I,J)$:P[At least one of N packets visiting (I,J) in h-hops]
  - Each packet is independent and identical

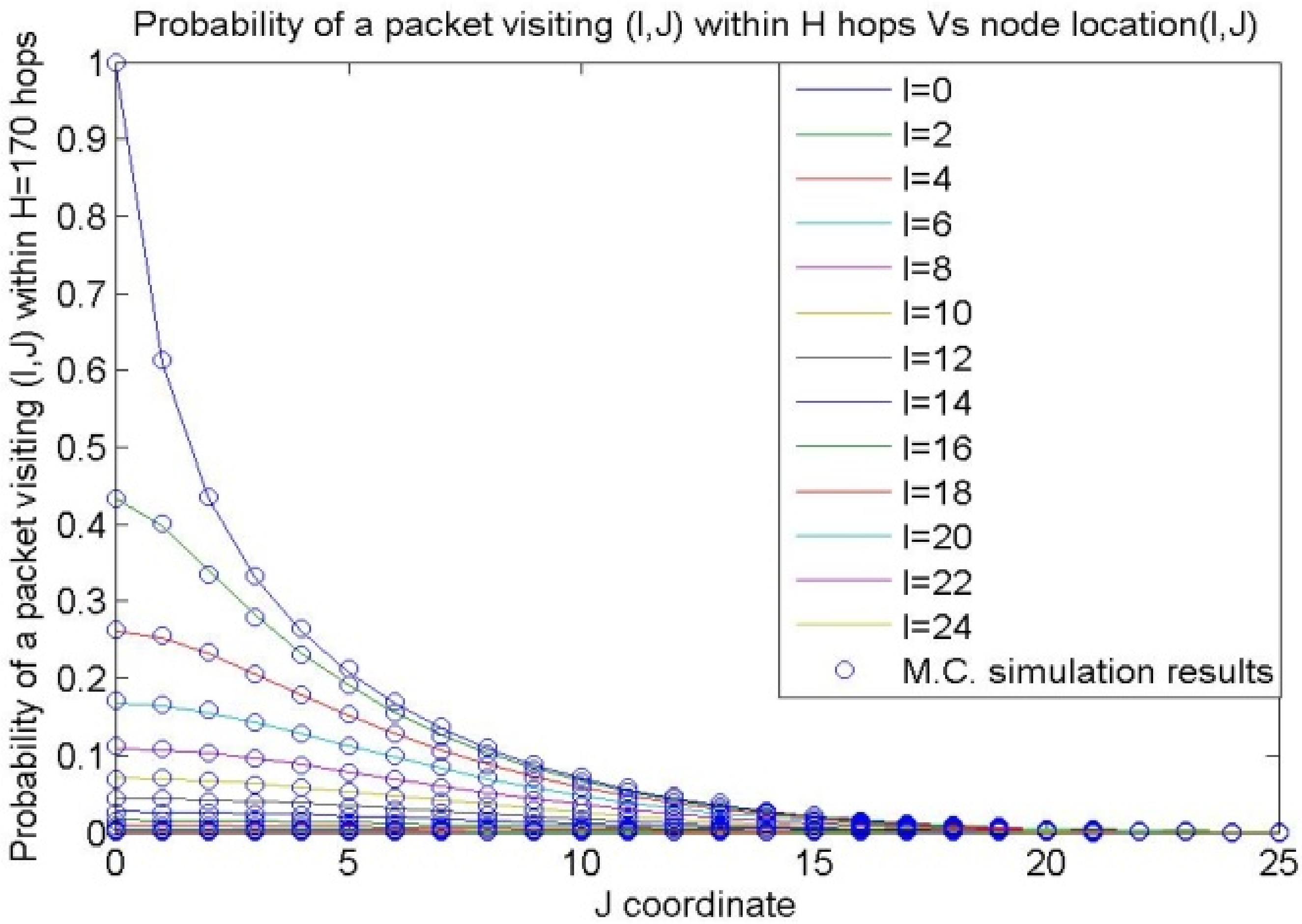
$$Q_h^{(N)}(I,J) = 1-(1-Q_h(I,J))^N$$

- P[Any of the N_e agents NOT meeting any of N_q queries]

$$M_{H_e,h_q}\Big|_{N_e,N_q} = \prod_{\forall (I+J)\le h_q}\left(1-Q_{H_e}{}^{(N_e)}(I,J)\,Q_{h_q}{}^{(N_q)}(I',J')\right)$$

Simulation Results

Case 01: Exact probability, $Q_H(I,J)$



Applications of the Model

Fixed energy budget

- Fixed packet length in lossless network
- Varying packet length in lossless network
- Fixed packet length in lossy network

Applications of the Model (Cntd.)

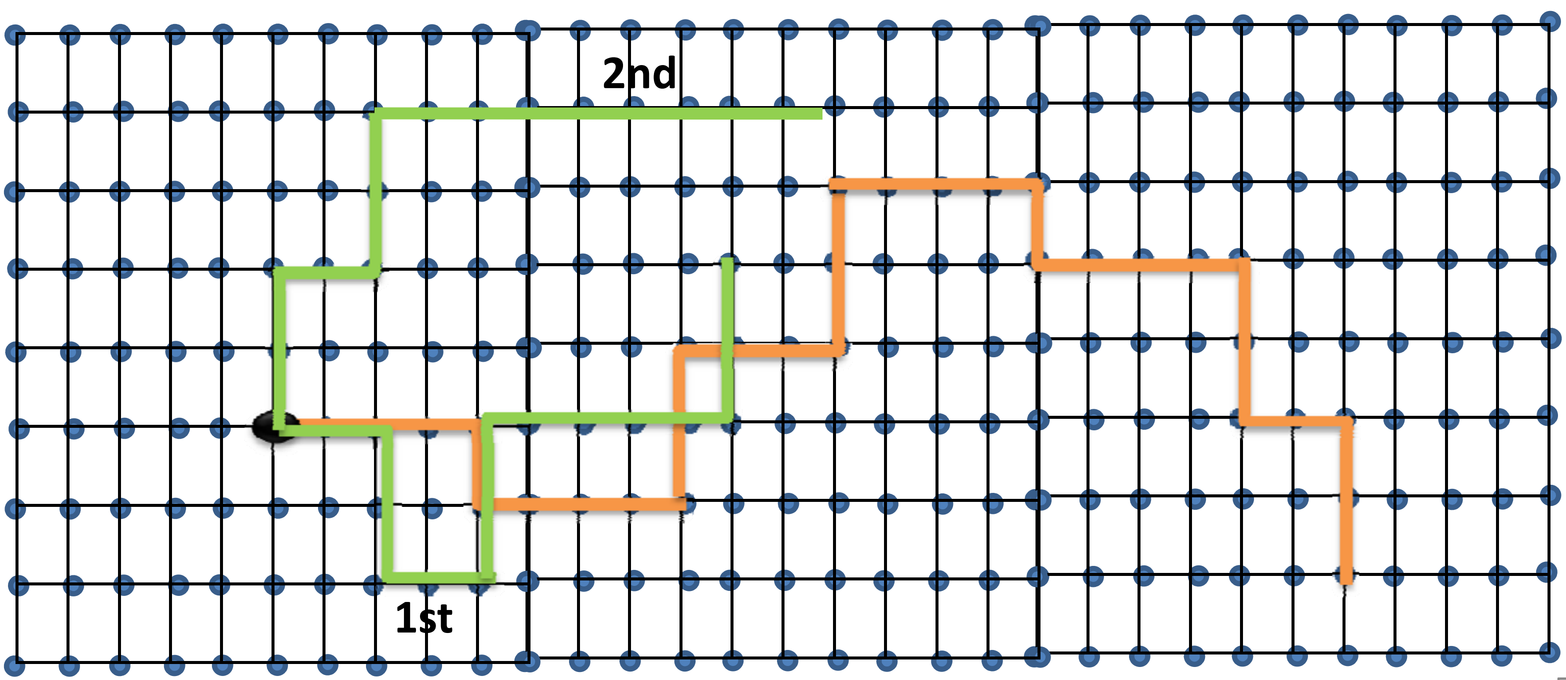
- No memory → Fixed packet length
- *Total energy consumption = Energy for agents*
  + Energy for queries
- Fixed total energy, i.e.
  - energy used for agents is fixed
  - energy used for queries is fixed

Applications of the Model (Cntd.)

Fixed total energy of agent(s)/query(s)

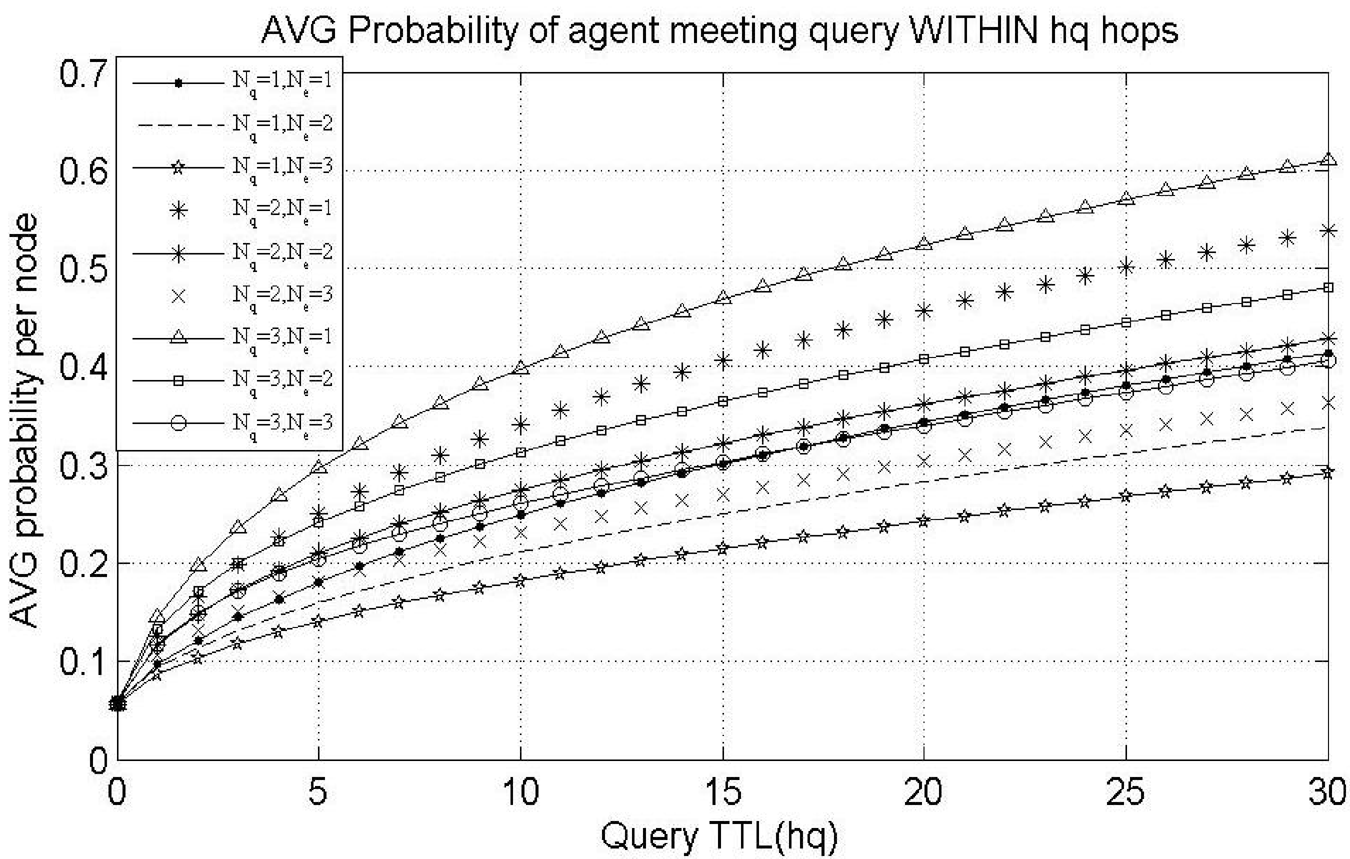
E.g.: 1 agent/query → TTL 30

2 agents/queries → TTL 15



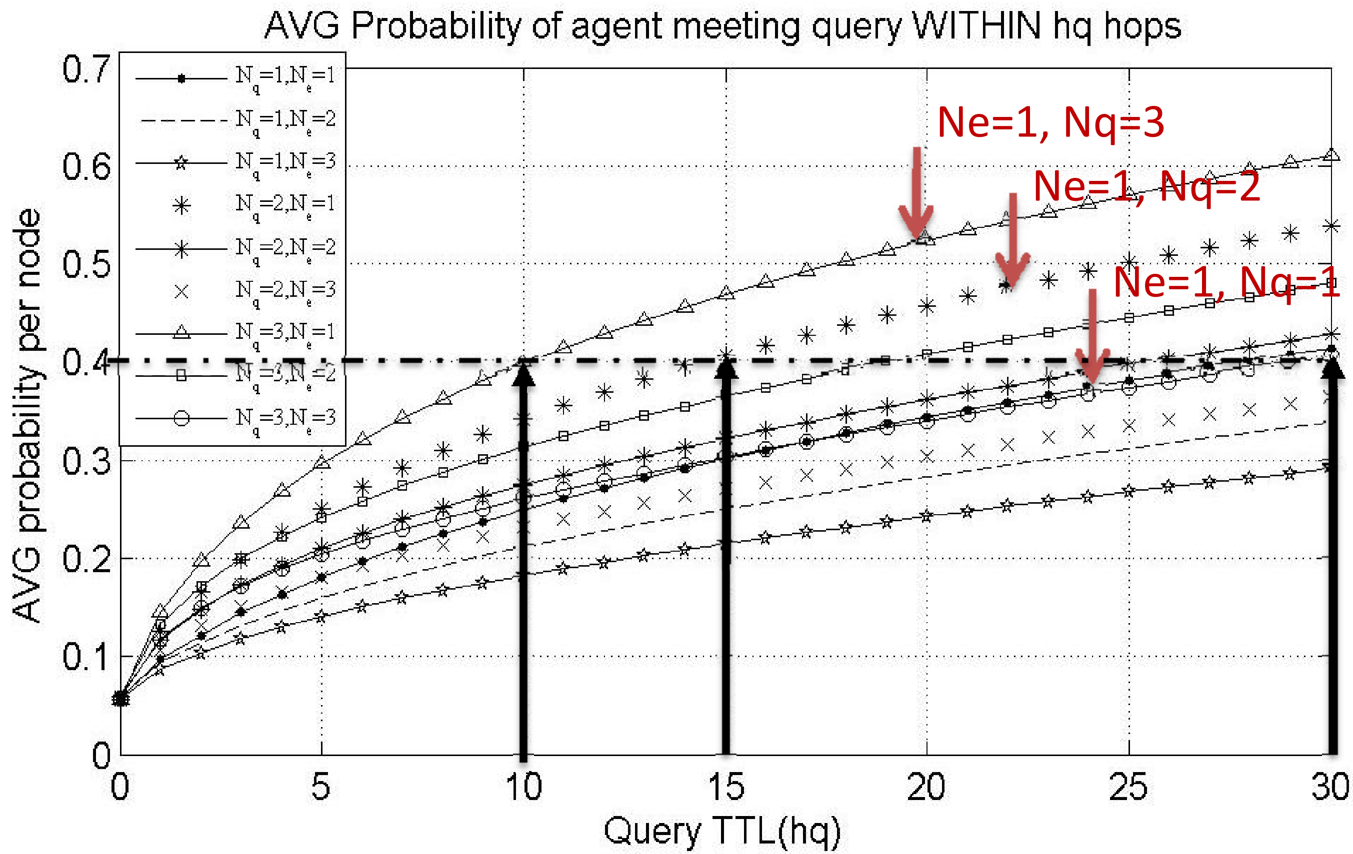
Applications of the Model (Cntd.)

Conclusion: Best performance is given by single agent under fixed agent energy constraint



Applications of the Model (Cntd.)

For query:



Conclusion: For a given energy allocation for queries, the reliability that can be achieved is independent of N_q

Summary and Discussion

- Issues → Optimal number of anchors required and improper anchor placements
- Properties of VCS
- Convex Subspace Routing
  - Improvements in routability and energy efficiency
  - Independent of anchor placement
  - No memory usage

Summary and Discussion(Cntd.)

- Derived the **exact** probabilities of
  - a packet visiting a node of interest within given number of hops
    - For 5 scenarios
  - agents meeting queries

  with Random Routing in rectangular grid
- Model can be used to select parameters for optimum performance
    - For 3 applications
- Model results hold even for sparse networks with node availability ≥ 75%

Future Work

- Improve the way of identifying convex routing surface
  - Virtual domain geometric relations
- Defining convex routing surface using more than 3 anchors
- How to identify **redundant** anchors
- Extend the Mathematical model to *n*-connected network
- Develop a virtual coordinate system using the past agents and queries in the network
  - Organized random routing

Thank you !

Questions

dulanjalie.dhanapala@colostate.edu