Beacon Vector Routing: Scalable Point-Point Routing in Wireless Sensornets

NSDI 2005

Speaker: Cheng-Han Wu

Outline

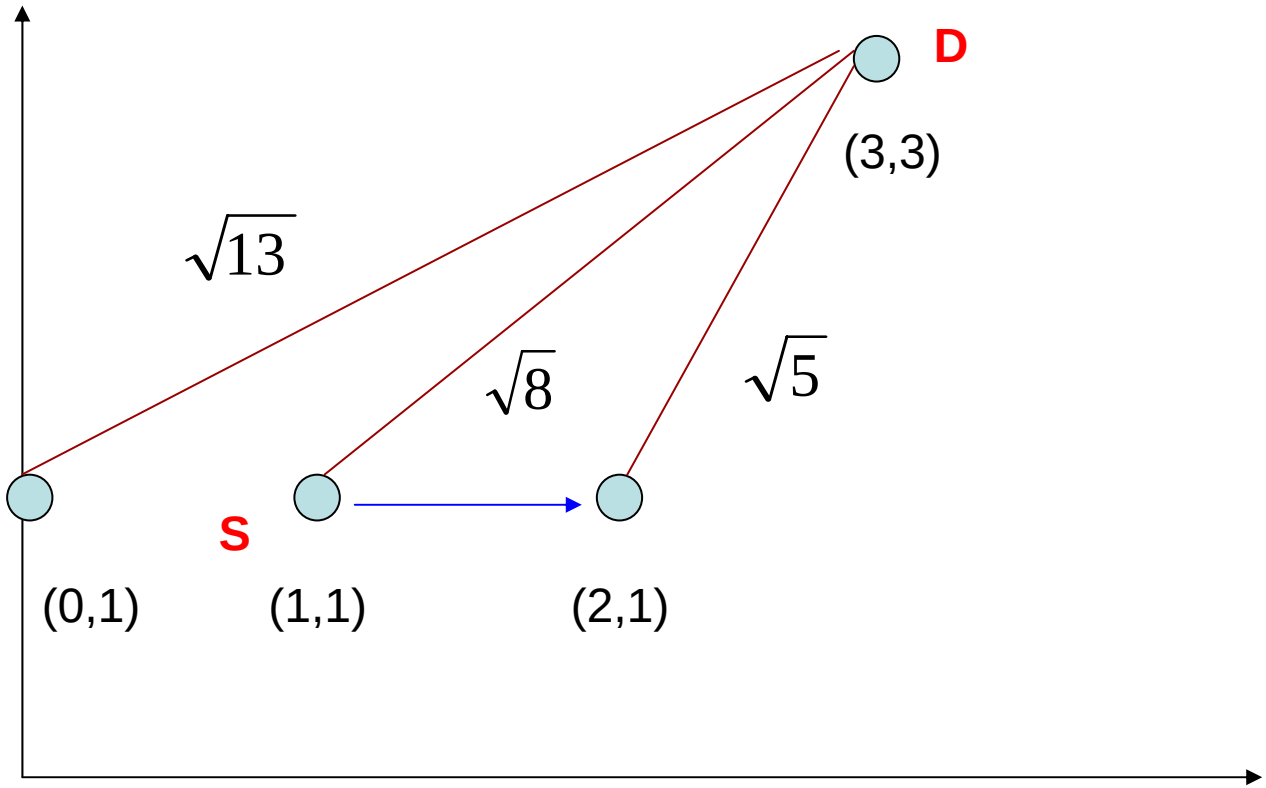
- Introduction
  - Geographic Routing

    -Physical Coordinates

    -Virtual Coordinates
- The BVR (Beacon Vector Routing) Algorithm
- Simulation Results
- Prototype Evaluation
- Conclusions

Introduction

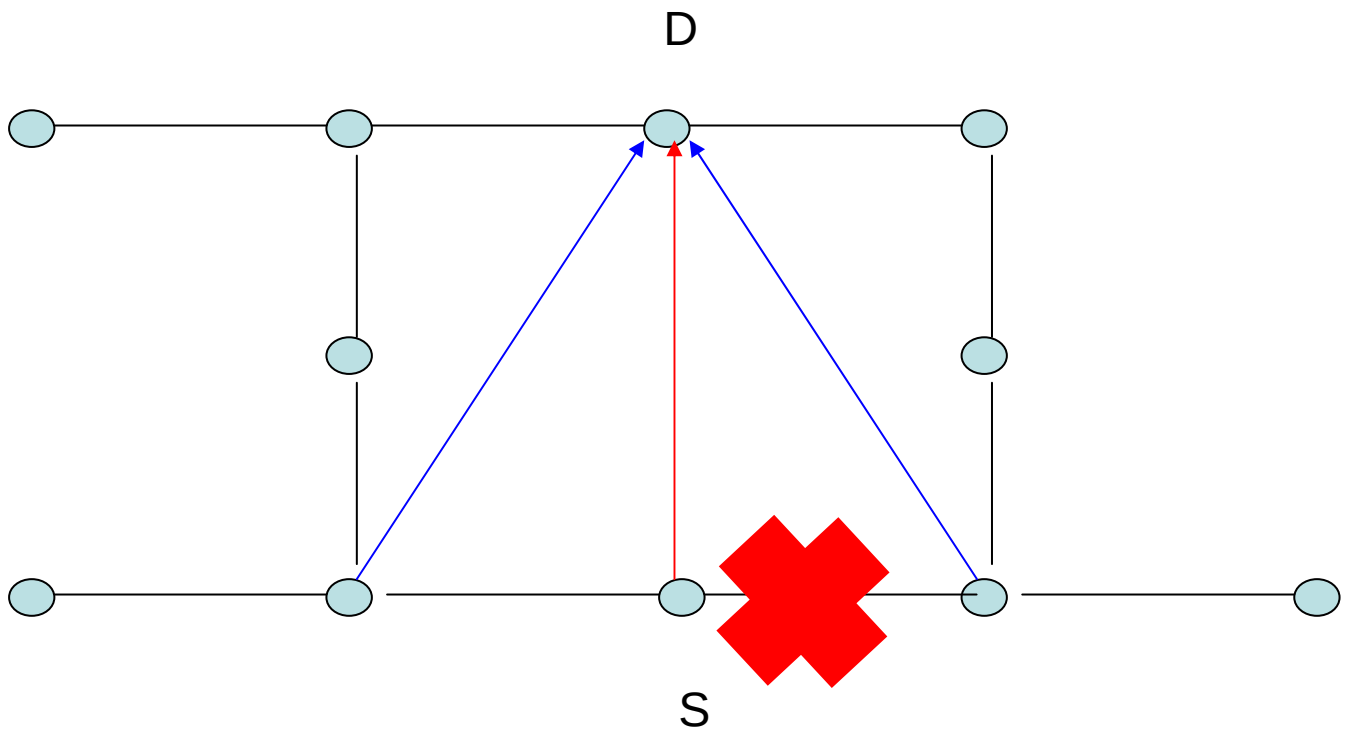
- Geographic routing



The Disadvantage of Physical Coordinates

- Sensors need to be equipped with GPS.
- It doesn’t work indoors.
- It is inefficient in low node density.

Why is Physical Coordinate Inefficient in Low Node Density?

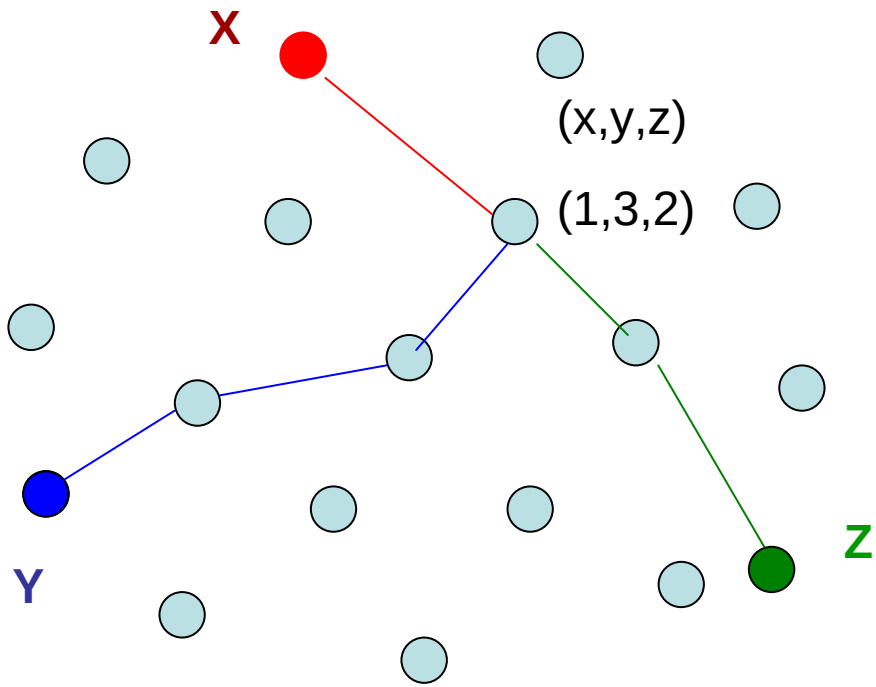


Why need virtual coordinate?

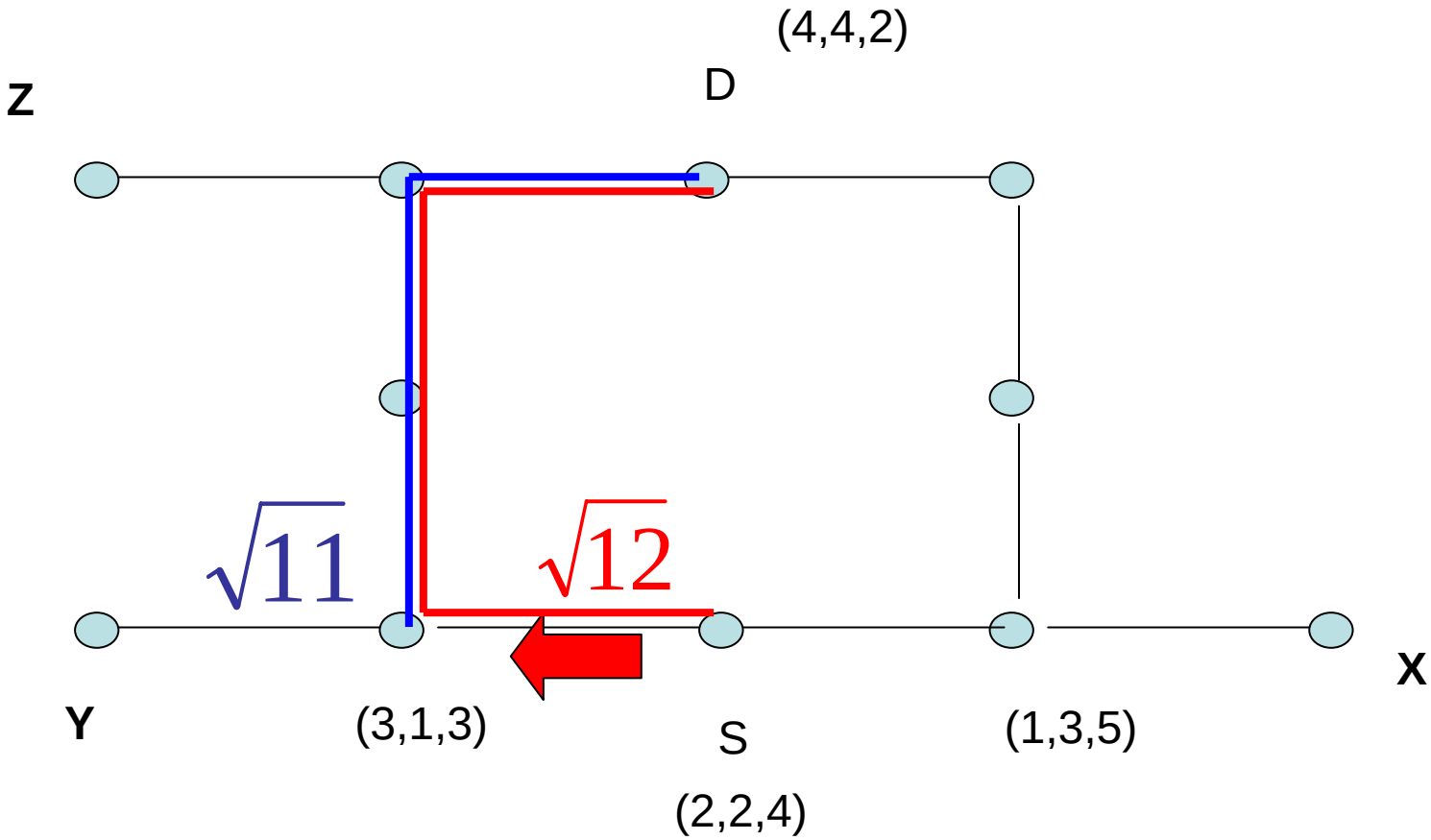
- It is cheaper than the physical coordinate.
- It can be used indoors.
- It is more inefficient than physical coordinate in low node density

The Virtual Coordinates

- GPS Free Coordinate Assignment and Routing in Wireless Sensor Networks. (INFOCOM)

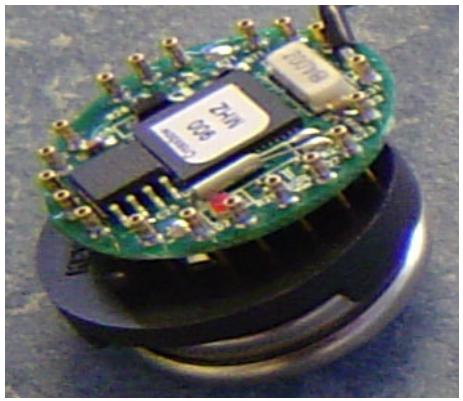


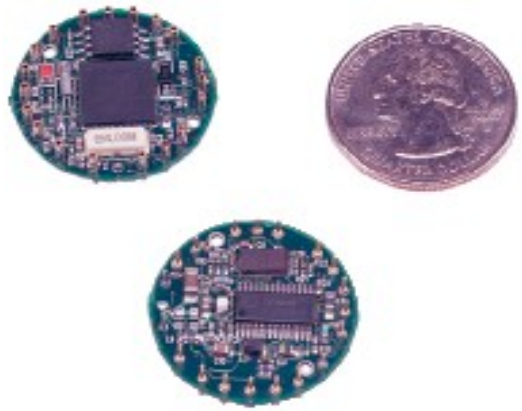
Virtual Coordinate is More Efficient than Physical Coordinate in Low Node Density



BVR: Beacon Vector Routing

- It requires very little state, overhead or pre-configured information(such as geographic).
- The BVR is implemented on the mica2dot motes.
- The mica2dot motes have several resource constraints- just 4KB of RAM, typical packet payloads of 29 bytes *etc.*





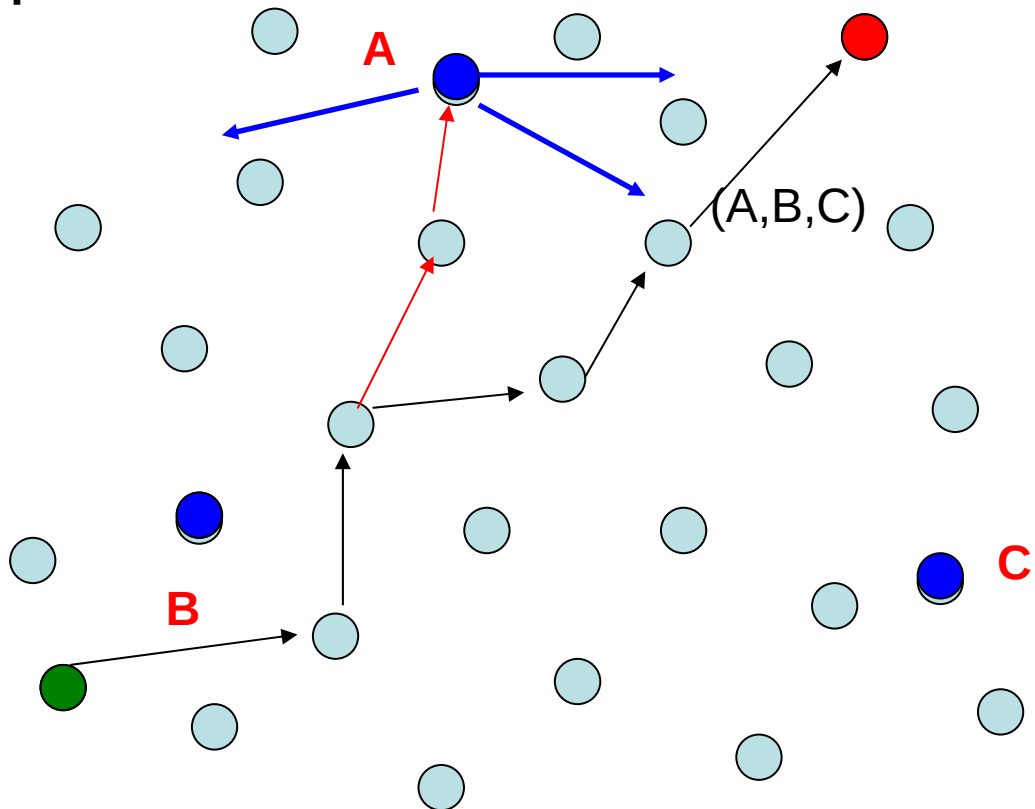
The BVR (Beacon Vector Routing) Algorithm

- **Construct the virtual coordinate**
- **Greedy forwarding over node coordinates**
- **If a node cannot make progress towards the destination by using greedy forwarding, it will forward to the beacon closest to the destination.**
- **A packet may ultimately reach the beacon closest to the destination and still not be able to make greedy progress. At this point, the root beacon initiates a scoped flood to find the destination.**

: beacon

:source

:destination

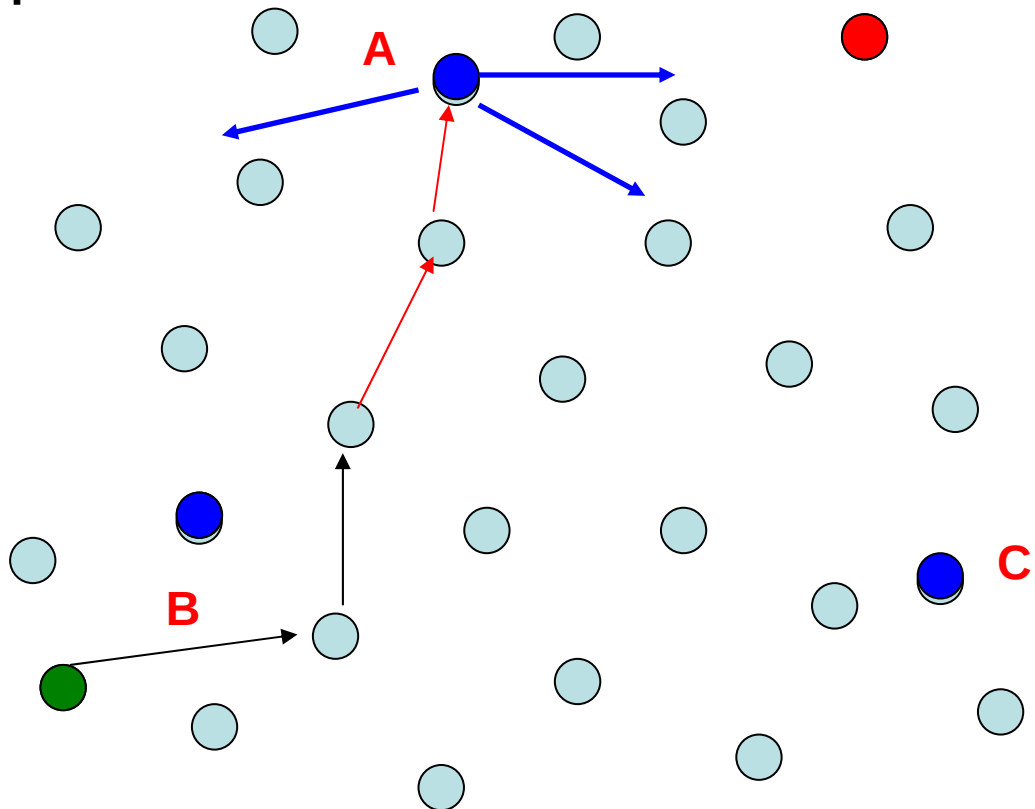


- **Construct the virtual coordinate**
- **Greedy forwarding over node coordinates**
- **If a node cannot make progress towards the destination by using greedy forwarding, it will forward to the beacon closest to the destination.**
- **A packet may ultimately reach the beacon closest to the destination and still not be able to make greedy progress. At this point, the root beacon initiates a scoped flood to find the destination.**

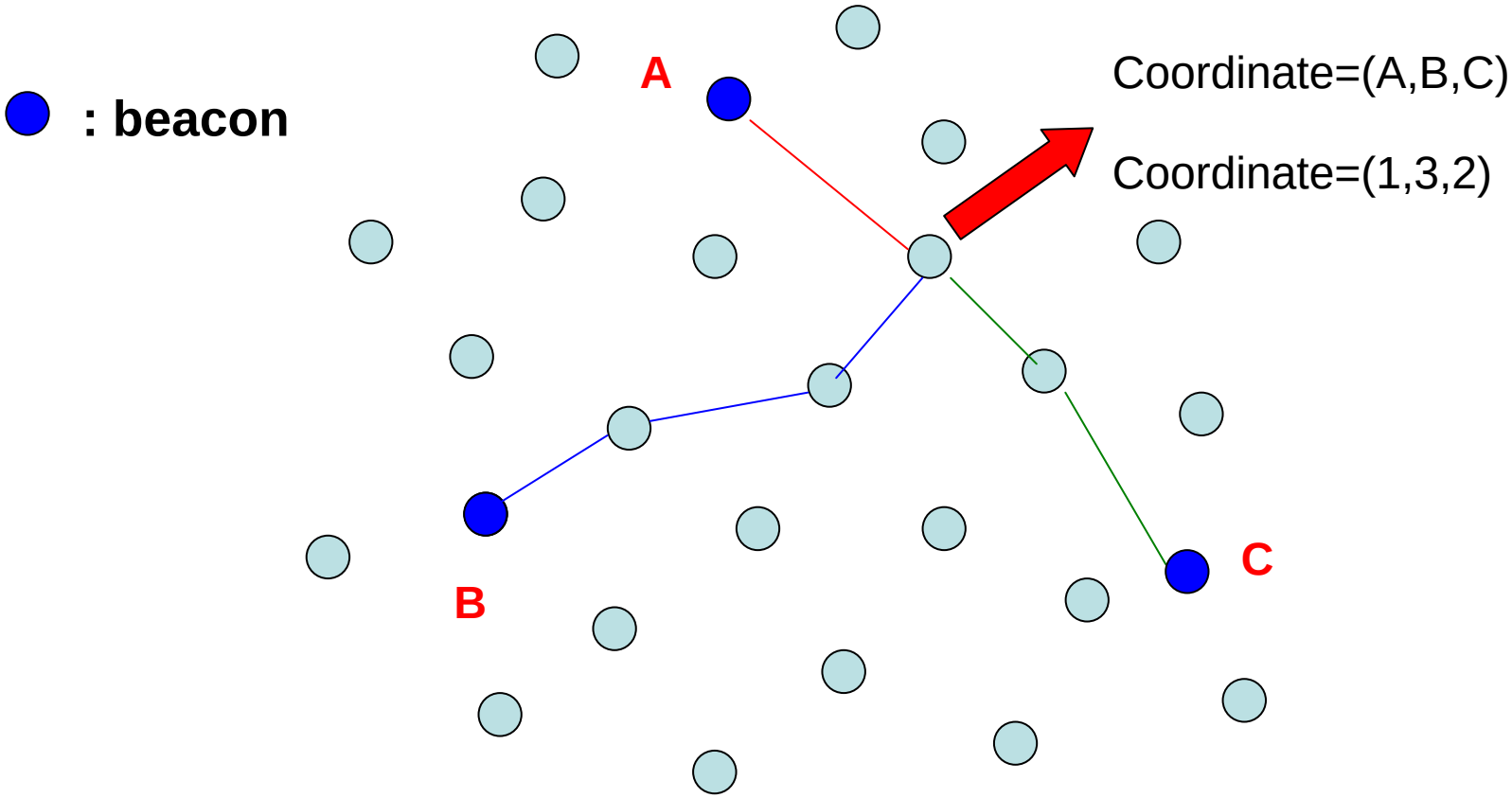
: beacon

:source

:destination



How to Construct the Virtual Coordinate?

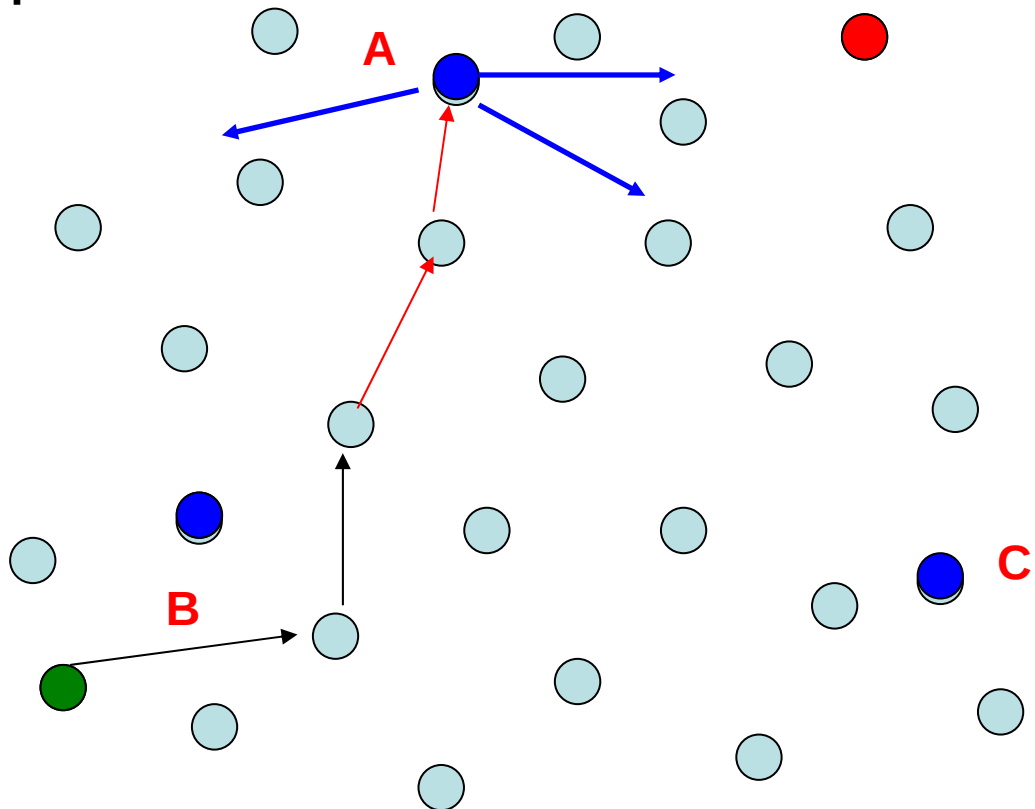


- **Construct the virtual coordinate**
- **Greedy forwarding over node coordinates**
- **If a node cannot make progress towards the destination by using greedy forwarding, it will forward to the beacon closest to the destination.**
- **A packet may ultimately reach the beacon closest to the destination and still not be able to make greedy progress. At this point, the root beacon initiates a scoped flood to find the destination.**

: beacon

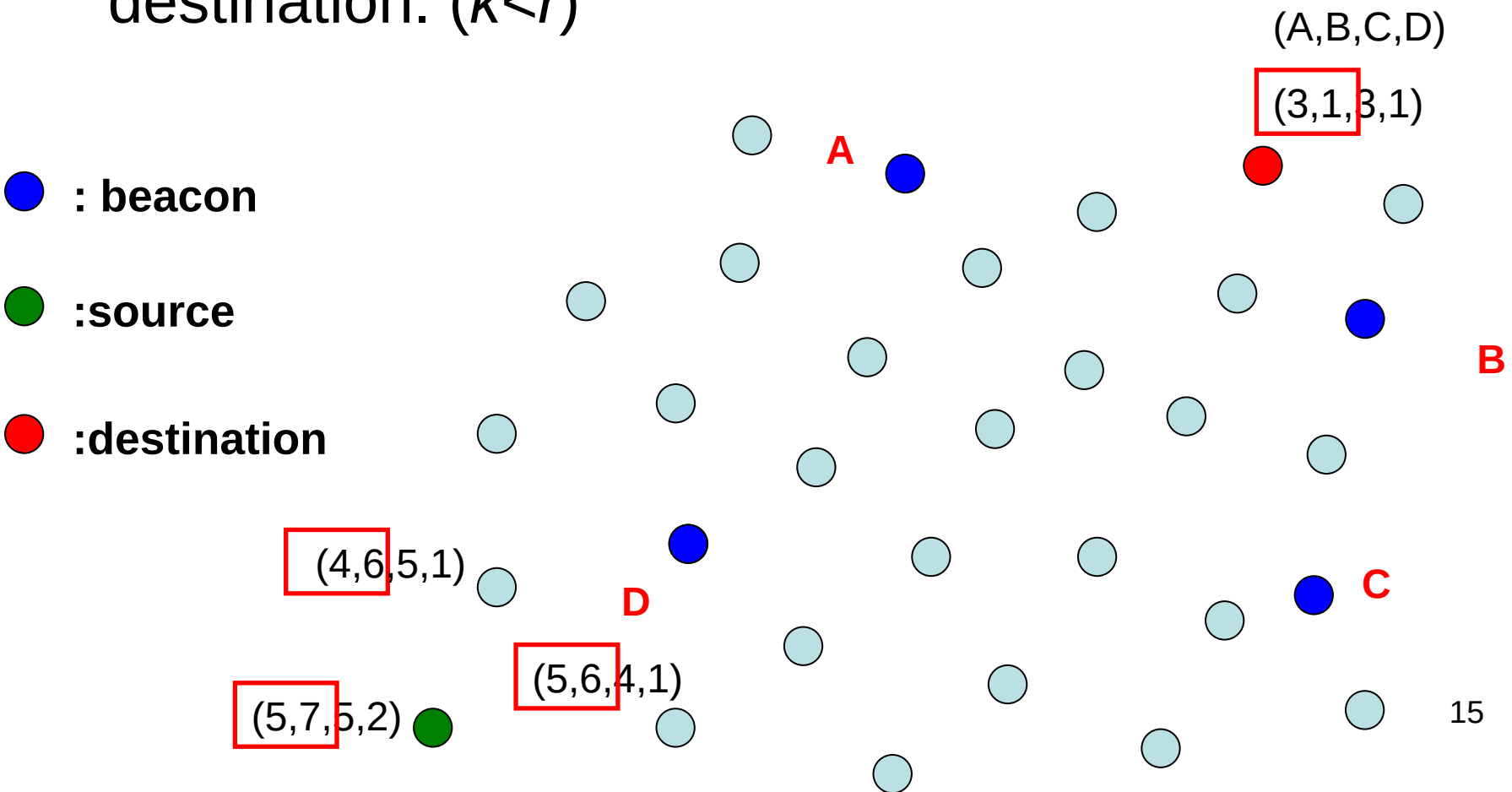
:source

:destination



Greedy Forwarding over Node Coordinates

- If there are *r* beacons, a node forwards the packet by only considering *k* closet beacons to destination. (*k*<*r*)



Greedy Forwarding over Node Coordinates

$$\delta_k^+(p,d) = \sum_{i \in C_k(d)} \max(p_i - d_i, 0) \quad \text{and} \quad \delta_k^-(p,d) = \sum_{i \in C_k(d)} \max(d_i - p_i, 0)$$

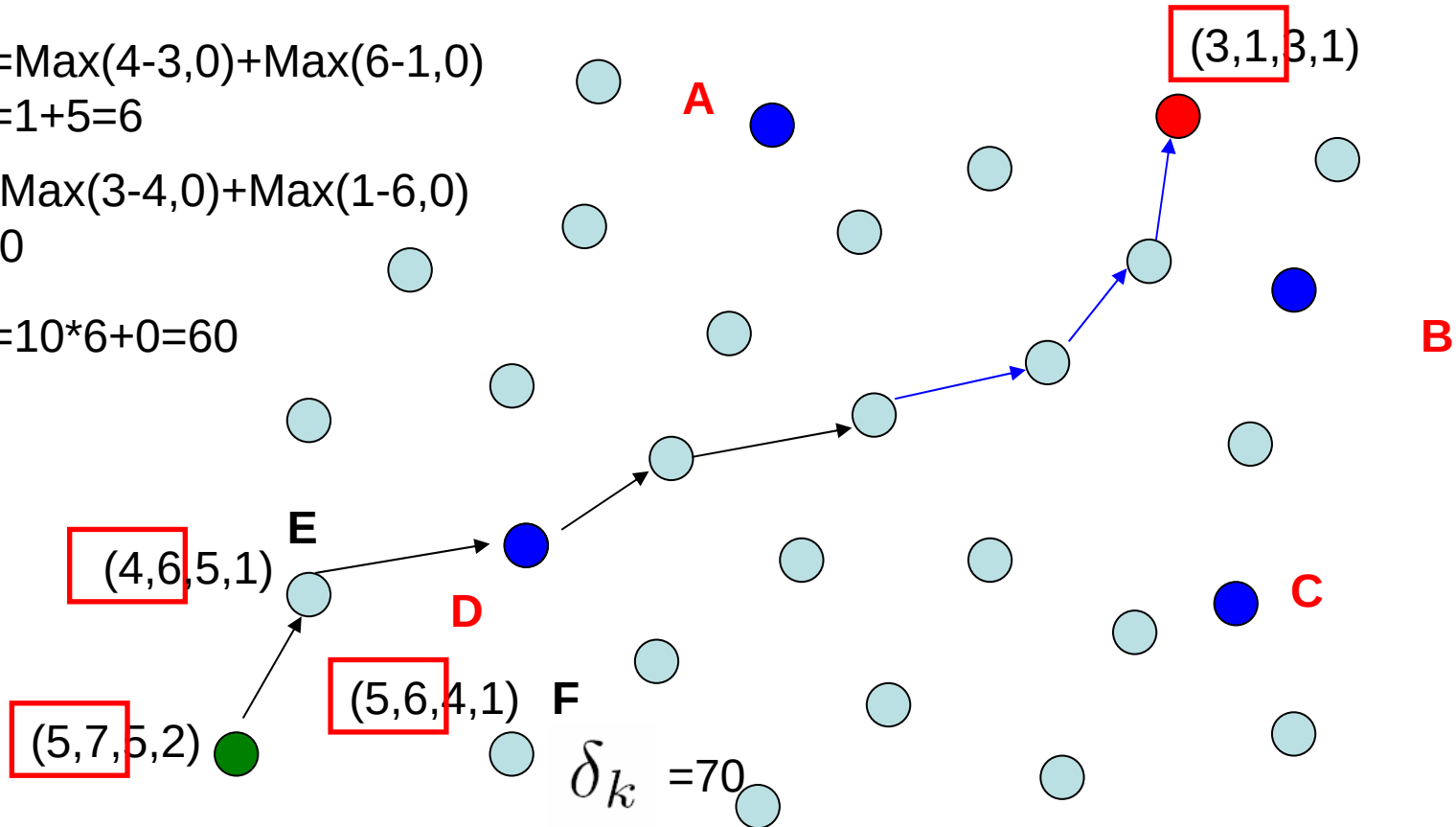
P: current routing node *d :* destination

$\delta_k = A\delta_k^+ + \delta_k^-$, A is constant (assume A=10)

δ_k^+ =Max(4-3,0)+Max(6-1,0)
=1+5=6

δ_k^- =Max(3-4,0)+Max(1-6,0)
=0

δ_k =10*6+0=60

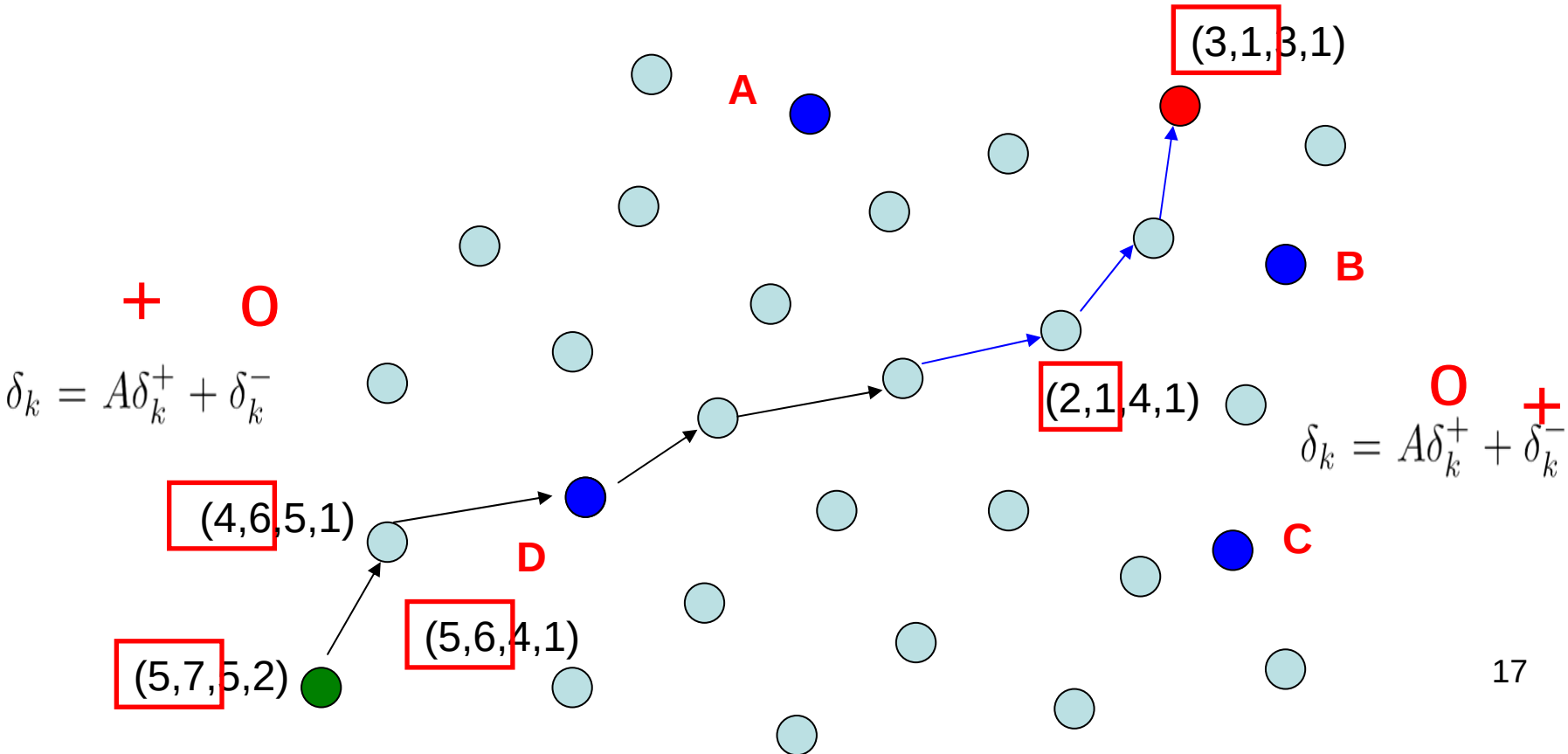


Greedy Forwarding over Node Coordinates

$$\delta_k^+(p,d) = \sum_{i \in C_k(d)} \max(p_i - d_i, 0) \quad \text{and} \quad \delta_k^-(p,d) = \sum_{i \in C_k(d)} \max(d_i - p_i, 0)$$

P: current routing node *d :* destination

$$\delta_k = A\delta_k^+ + \delta_k^-$$, A is constant (assume A=10)

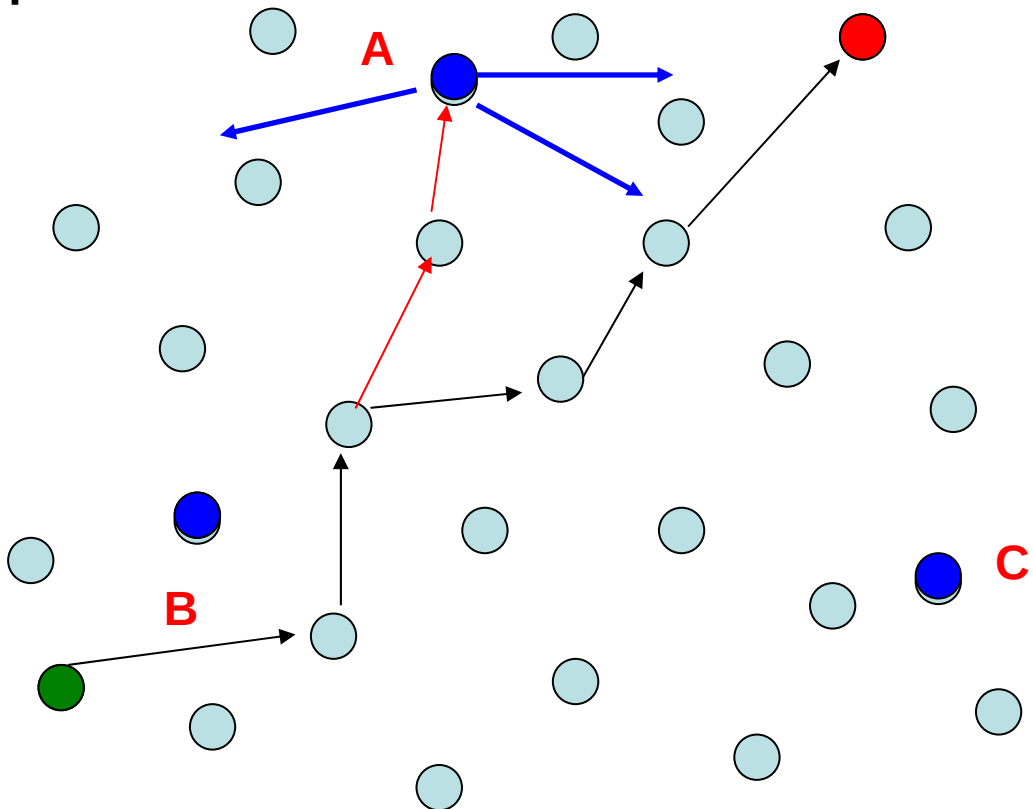


- **Construct the virtual coordinate**
- **Greedy forwarding over node coordinates**
- **If a node cannot make progress towards the destination by using greedy forwarding, it will forward to the beacon closest to the destination.**
- **A packet may ultimately reach the beacon closest to the destination and still not be able to make greedy progress. At this point, the root beacon initiates a scoped flood to find the destination.**

: beacon

:source

:destination



Simulation Results

- Assumptions:
  - Each sensor has a fixed circular radio range
  - A node can communicate with all and only those nodes that fall within its range.
  - The simulator ignores the capacity of, and congestion in the network.
  - The simulator ignores packet losses.

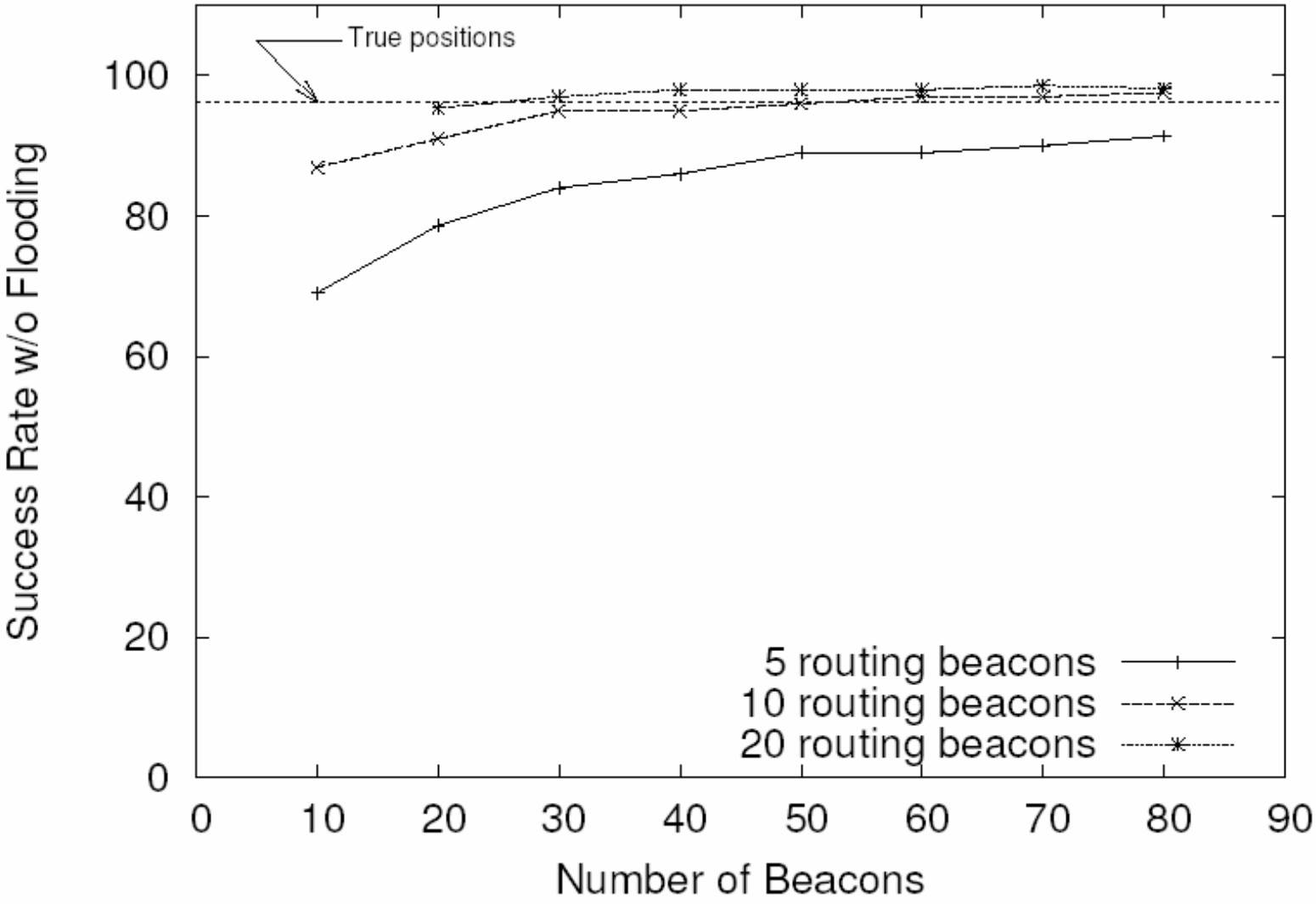
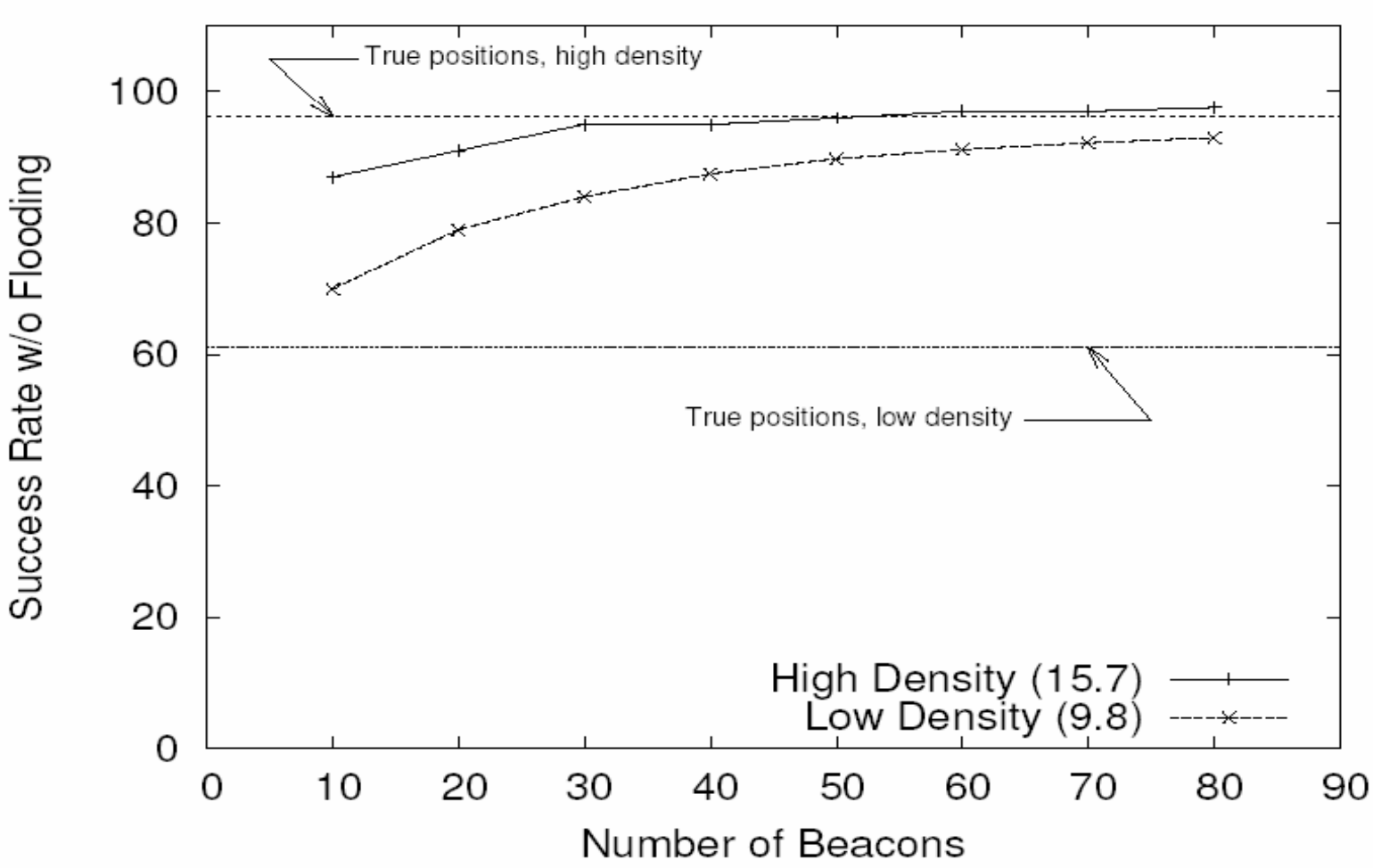
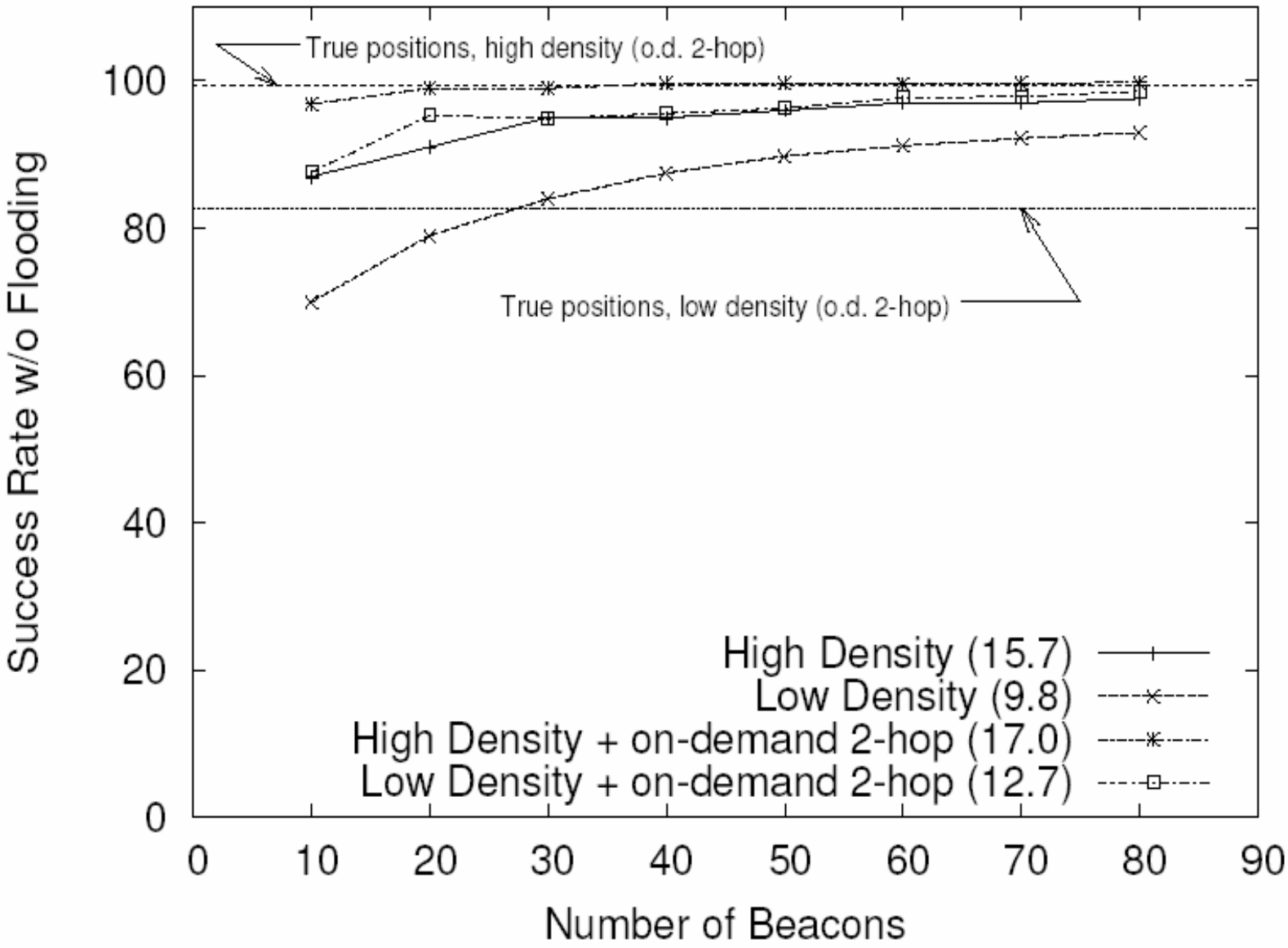


Figure 1: Success rate of routes without flooding in a 3200 node network, for different numbers of total beacons, r, and routing beacons, k.

The Impact of Node Density



The Impact of Node Density



How many beacons do we need?

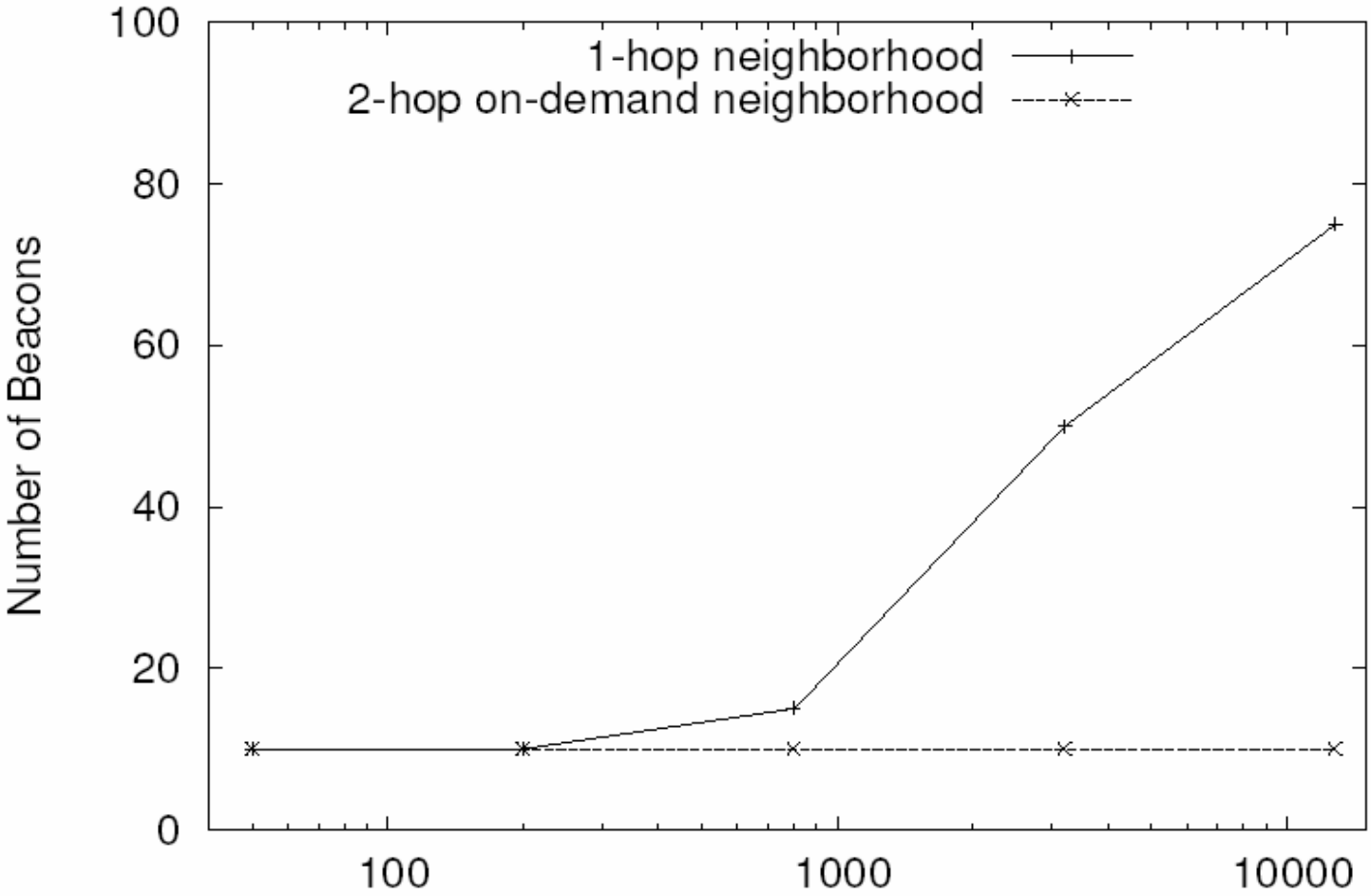
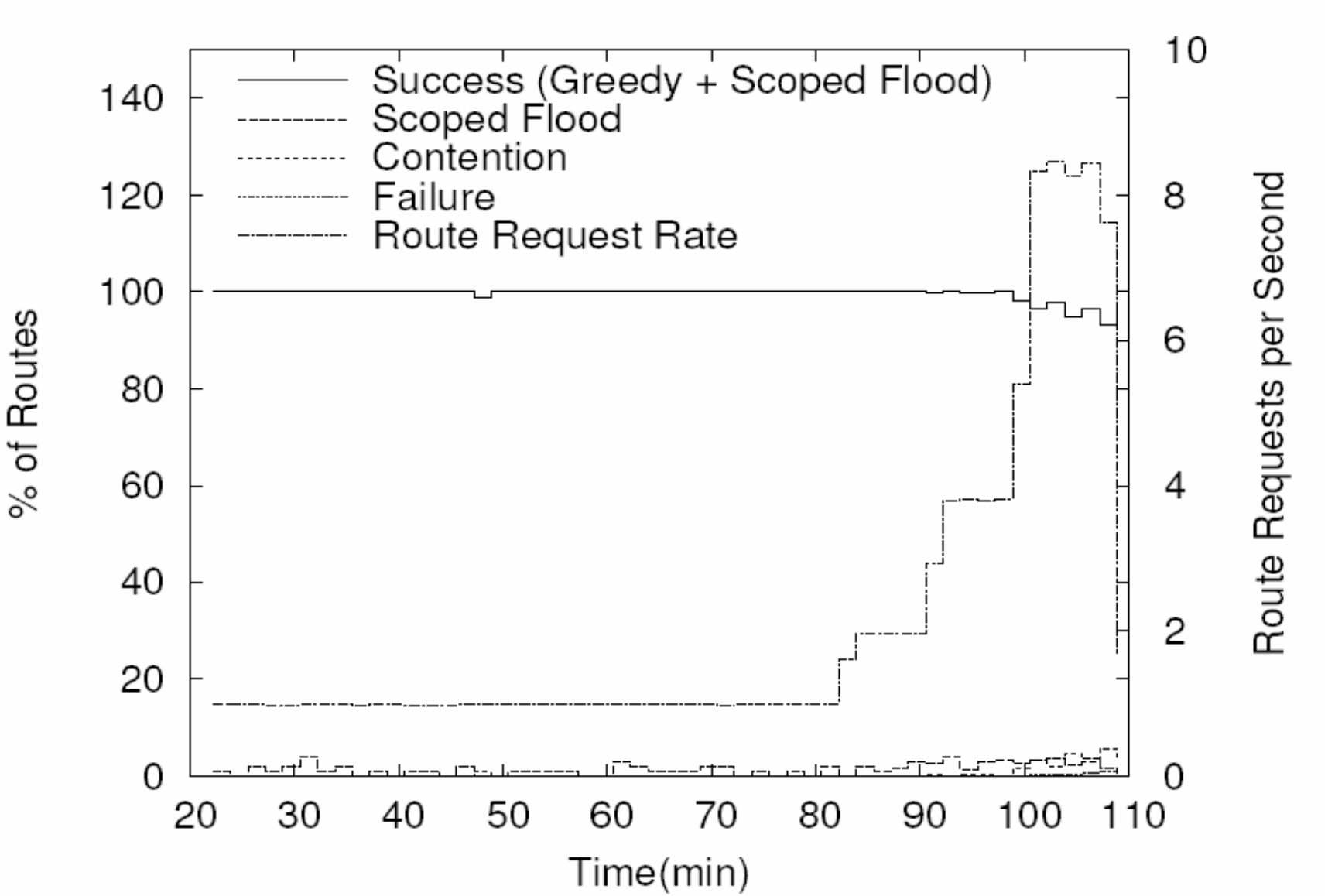


Figure 4: Number of beacons required to achieve less than 5% of scoped floods, with $k = 10$ routing beacons.

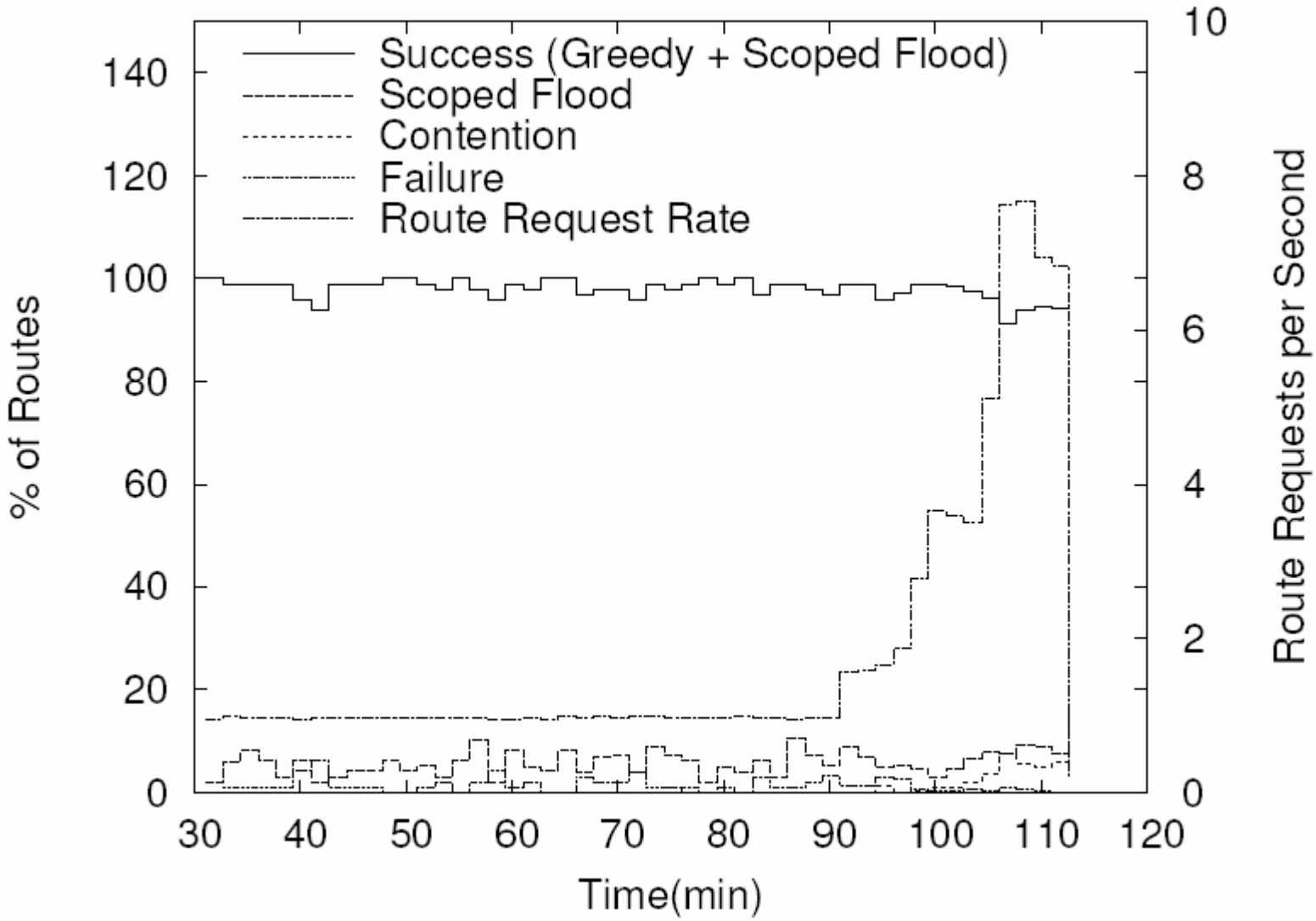
Prototype Evaluation

- Two testbeds:
  - (Office-Net) consists of 42 mica2dot motes in an indoor office environment of approximately 20X50m.
  - (Univ-Net) is a testbed of about 74 mica2dot motes deployed across multiple student offices on a single floor of UC Berkeley's Computer Science building.

Office-Net


Success (Greedy + Scoped Flood)
Scoped Flood
Contention
Failure
Route Request Rate
% of Routes
Route Requests per Second
Time(min)
0
20
40
60
80
100
120
140
2
4
6
8
10
20
30
40
50
60
70
80
90
100
110

Univ-Net


Success (Greedy + Scoped Flood)
Scoped Flood
Contention
Failure
Route Request Rate
% of Routes
Route Requests per Second
Time(min)
0
20
40
60
80
100
120
140
0
2
4
6
8
10
30
40
50
60
70
80
90
100
110
120

Conclusions

- Beacon Vector Routing is a new approach to achieving scalable point-to-point routing in wireless sensornets.

- The advantage of BVR are its simplicity ,making it easy to implement on resource constrained nodes like motes.