ALBA: a cross-layer integrated protocol stack for medium-large scale wireless sensor networks

Michele Nati, Chiara Petrioli[†]

[†] *Department of Computer Science – University of Rome "Sapienza" – Italy*

Authors: P. Casari, M. Nati, C. Petrioli, M. Zorzi

This work was partially supported by the European 6th Framework Programme through the Integrated Project (IP) ***e-SENSE****, contract number 027227.*

Outline

- Geographic routing concepts
- Handling dead ends: Related work
- Adaptive Load-Balancing Algorithm (ALBA)
- Rainbow
  - A node-coloring algorithm to route around dead ends
- Simulations settings
- Results for high and low nodal densities
- Impact of localization errors
- Conclusions and discussion

The geographic routing paradigm

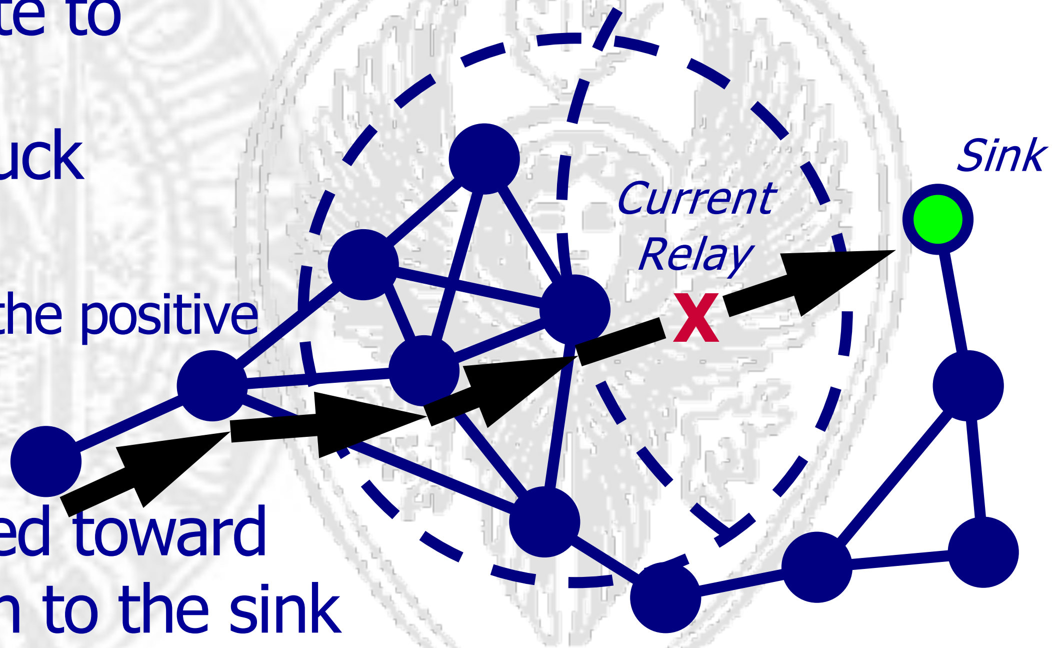
Geographic routing

"Forward the packet to a node that offers geographic advancement toward the destination"

- Pros
  - Virtually *stateless* (needs only knowledge of the source's and the destination's locations)
- Cons
  - Requires positioning estimation (BUT is it really critical?)
  - Requires mechanisms to route packets out of *dead ends*
    - ✓ The present relay is the closest to, yet not a neighbor of, the destination

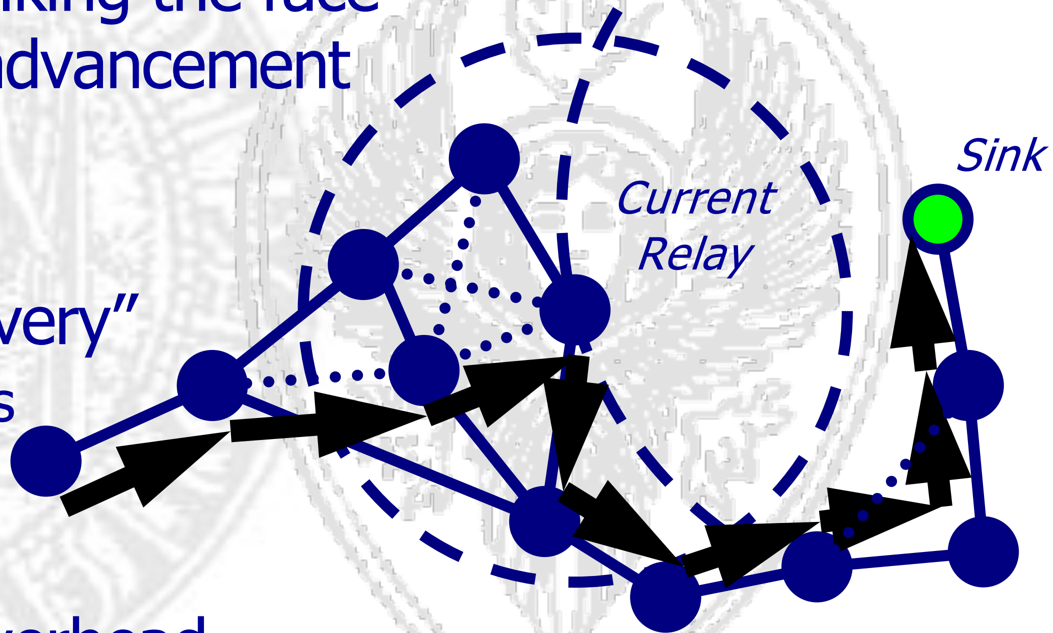
The dead end problem

- If the routing algorithm is tuned to achieve a positive advancement at each step, dead ends may occur
- In this example, a route to the sink is available
  but the packets get stuck
  at the current relay
  - There are no nodes in the positive advancement area
- Packet losses occur if data are not re-routed toward nodes that have a path to the sink



The dead end problem, 2

- Current approaches to dead end resolution include **planarizing** the network graph (the resulting graph has no cross links) and walking the face perimeters when the advancement area is empty

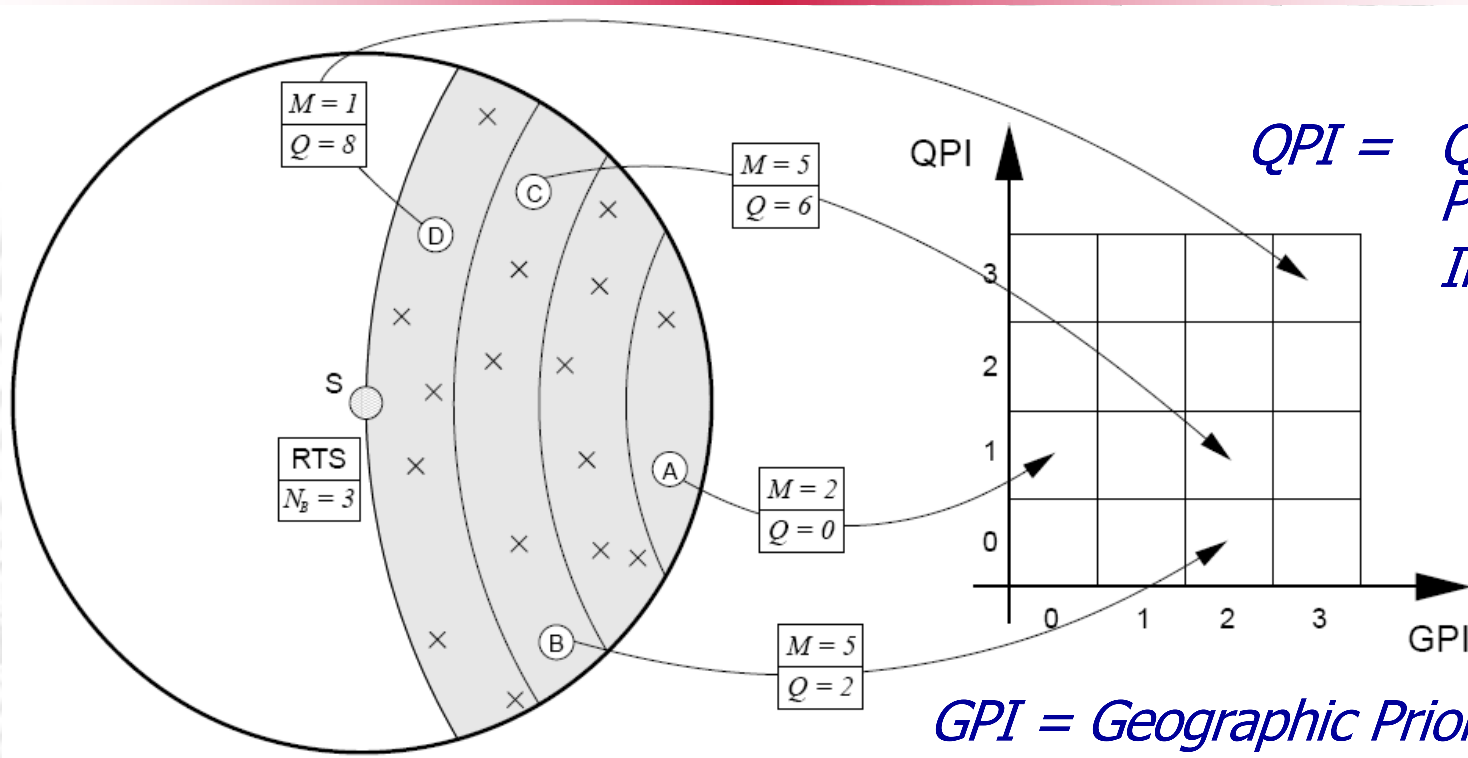


- **Pros**: "Guarantee delivery"
  - Planarization algorithms can be distributed
- **Cons**: planarization overhead, prone to location and channel errors

Our Approach: Basics

- ALBA → Adaptive Load-Balancing Algorithm
  - Integrates interest dissemination and converge-casting
  - Cross layer optimized converge-casting
    - ✓ MAC
    - ✓ Geographic Routing
    - ✓ Mechanisms to load balance traffic among nodes (to decrease the data funneling effect)
    - ✓ Schemes to distributely and efficiently deal with dead ends
- Operations:
  - Nodes forward packets in bursts (up to M_B packets sent back-to-back)
    - ✓ The length of the burst is adapted
  - Forwarders are elected based on
    - ✓ The ability to receive and correctly forward packets
      - The used metric involves the queue level, the past transmission history of the relay, and the number of packets the sender needs to transmit
    - ✓ The geographic proximity to the destination

Our Approach: Basics of the ALBA Protocol



QPI = Queue Priority Index

GPI = Geographic Priority Index

$$QPI = \lceil (Q + N_B) / M \rceil - 1$$

Queue level

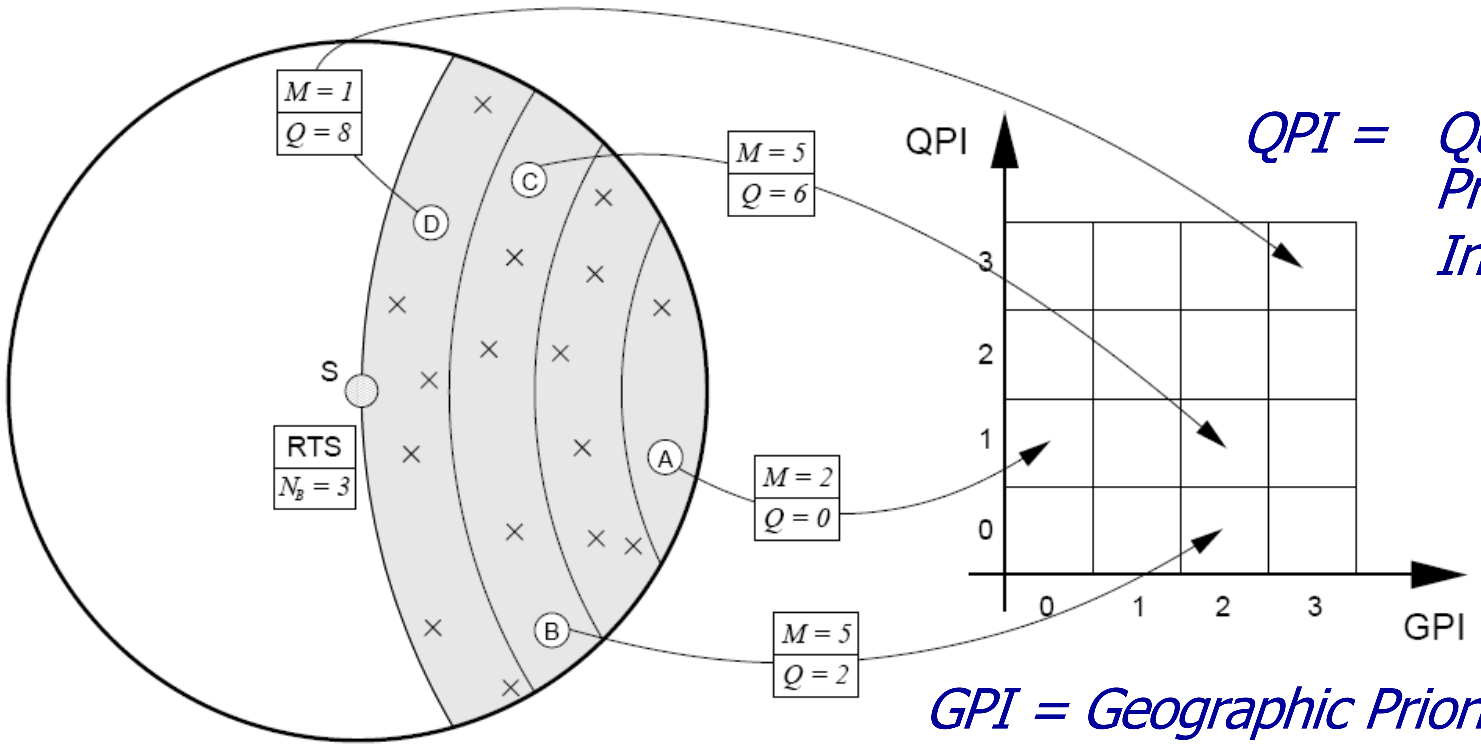
Requested length of the burst

Average length of a burst the relay expects to transmit correctly

ALBA Features

- The metric used for the choice of the relay ensures load balancing as it preferably chooses relays with
  - Low queue, especially if N_B is high
  - Good forwarding history (through *M*)
- Nodes employ duty-cycling to enforce energy saving
- The relay selection works in phases
  - Phase 1: Selection of the best QPI
    - ✓ Attempt 1 search for QPI=0, Attempt 2 for QPI=0,1, and so on
    - ✓ **Awaking nodes can participate in this selection phase**
  - Phase 2: Selection of the best GPI
    - ✓ Performed if more than one node with the same QPI was found
    - ✓ Awaking nodes cannot participate here (to speed up completion)
- **Still prone to dead ends**

ALBA: An example

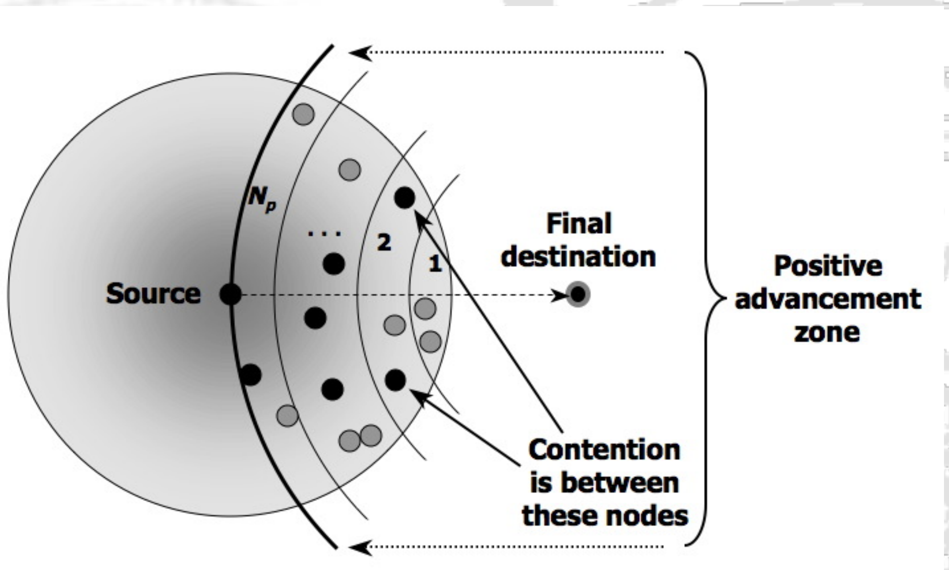


QPI = Queue Priority Index

GPI = Geographic Priority Index

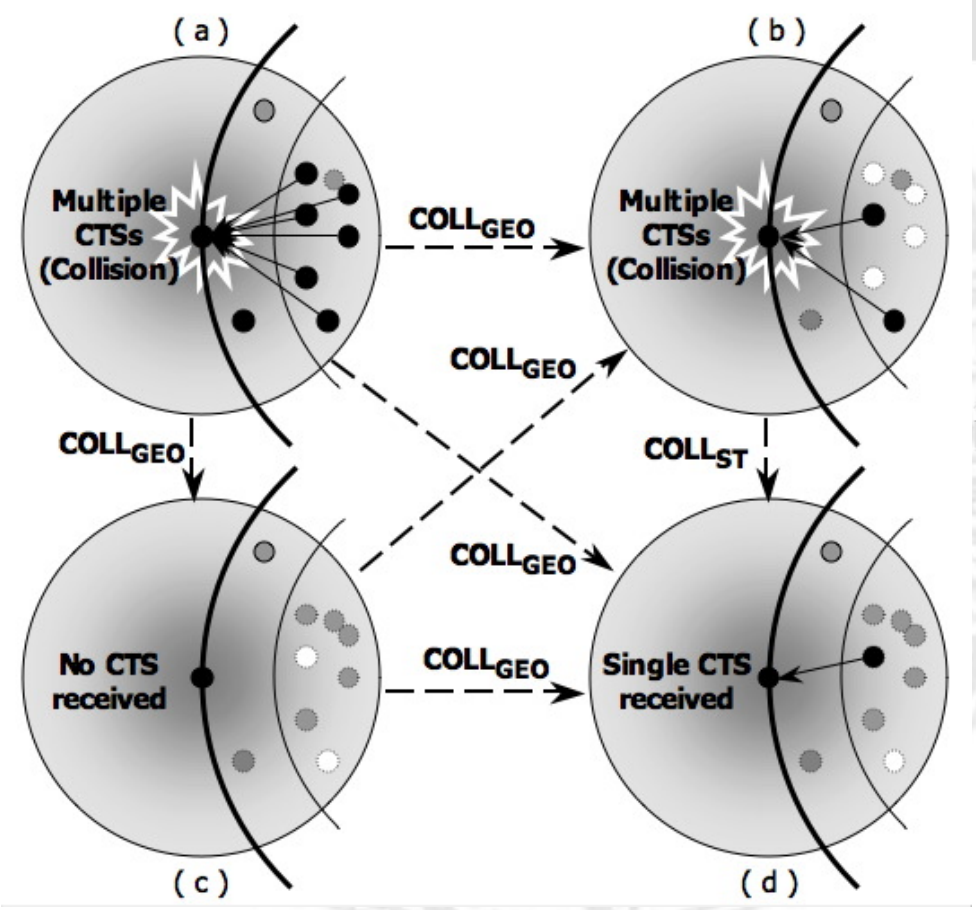
1. Node A is nearer to the sink (GPI =1) but has a low QPI (M=2); node B, is farther but has greater reliability (M) and comparable queue occupancy (Q); B has a greater QPI than A
2. In case of node B is sleeping at transmission time, node A is selected for its better GPI

Contention Mechanism



- Source nodes send a RTS msg to query relays. Relays respond with CTSs
- No response: a CONTINUE msg pings the following region
- Collision: a COLLISION msg starts the collision resolution algorithm
- CTS received: Burst of DATA transmission starts

Collision Resolution Algorithm



- Upon receiving a COLL$_{GEO}$ msg, nodes reply based on their GPI
- Upon receiving a COLL$_{ST}$ msg, nodes persist in sending CTSs with probability 0.5
- Should they all decide to stay silent, the following COLL$_{ST}$ msg enables a further decision
- Eventually, the process ends with a single valid relay being selected

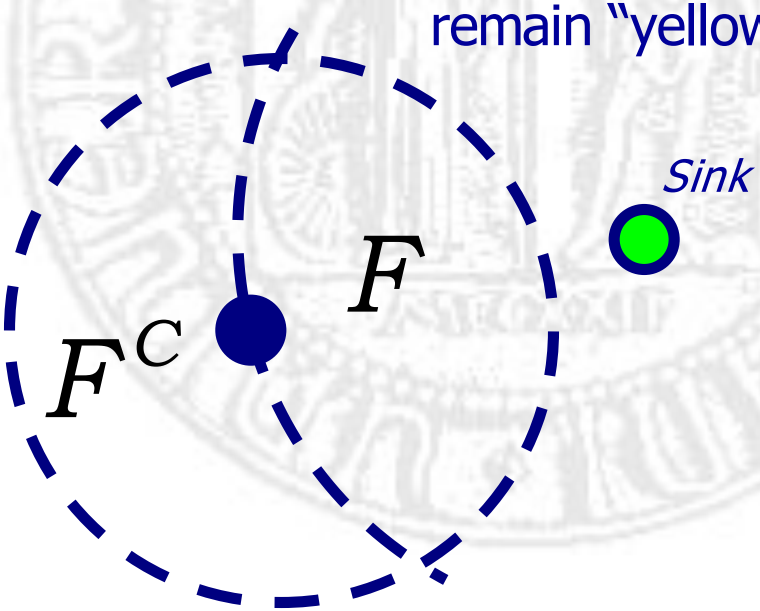
Rainbow

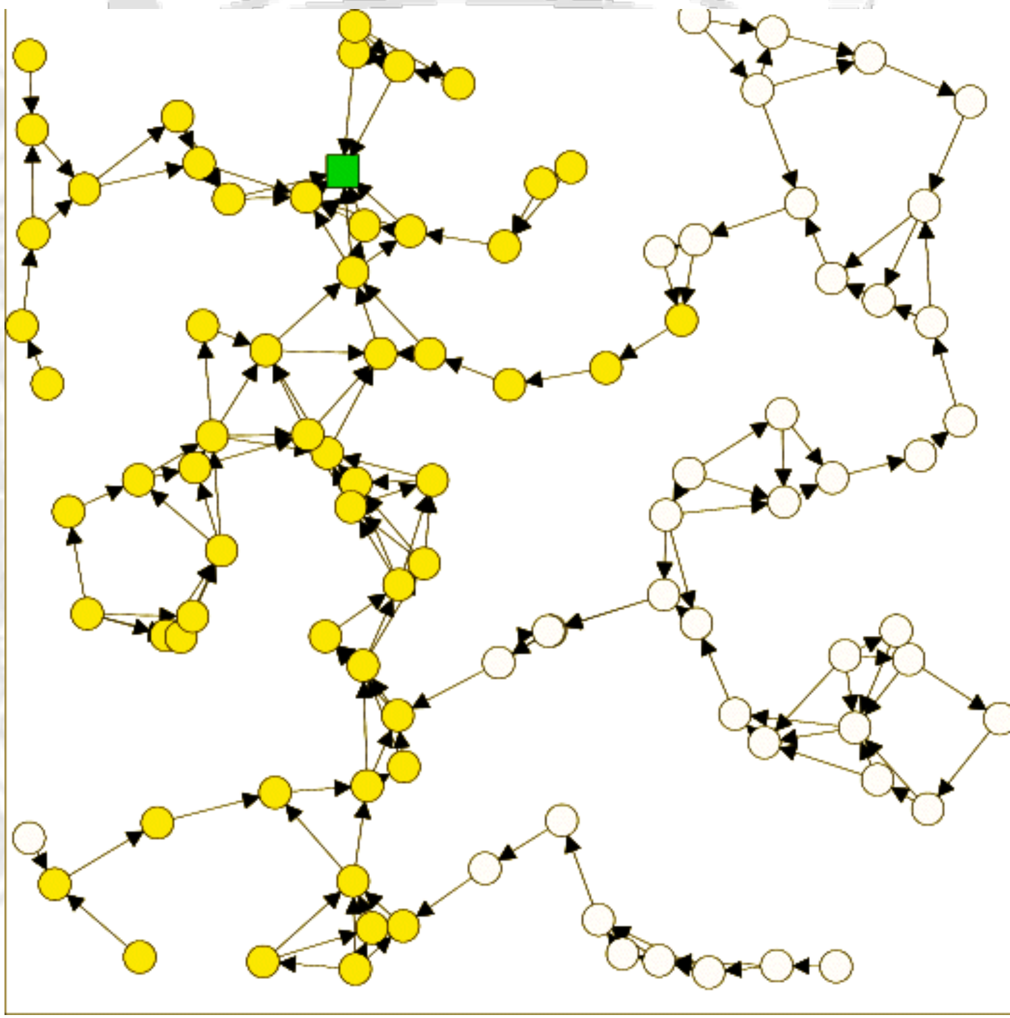
A node coloring algorithm for routing out of dead ends and around connectivity holes

- Concepts
  - In low density topologies, a method for routing around dead ends is needed
  - Nodes that recognize themselves as dead ends progressively stop volunteering as relays
  - To route traffic out of the dead end, they begin to transmit packets backward, in the negative advancement zone
  - Hopefully, a relay that has a greedy forwarding path to the sink can eventually be found
  - A recursive coloring procedure is used

Rainbow node coloring scheme – Yellow nodes

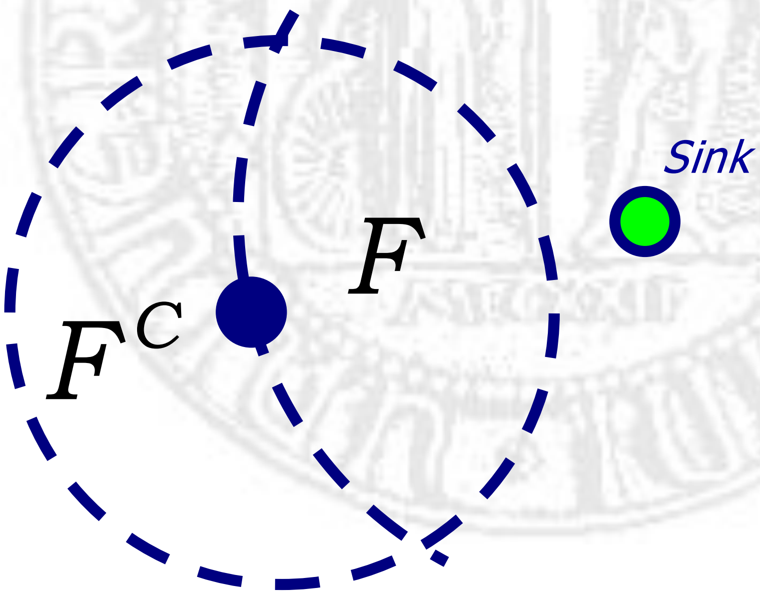
- F and F^C are the positive and negative advancement areas, respectively
- Initially, all nodes are "yellow"
- All nodes that exhibit a greedy path to the sink remain "yellow"

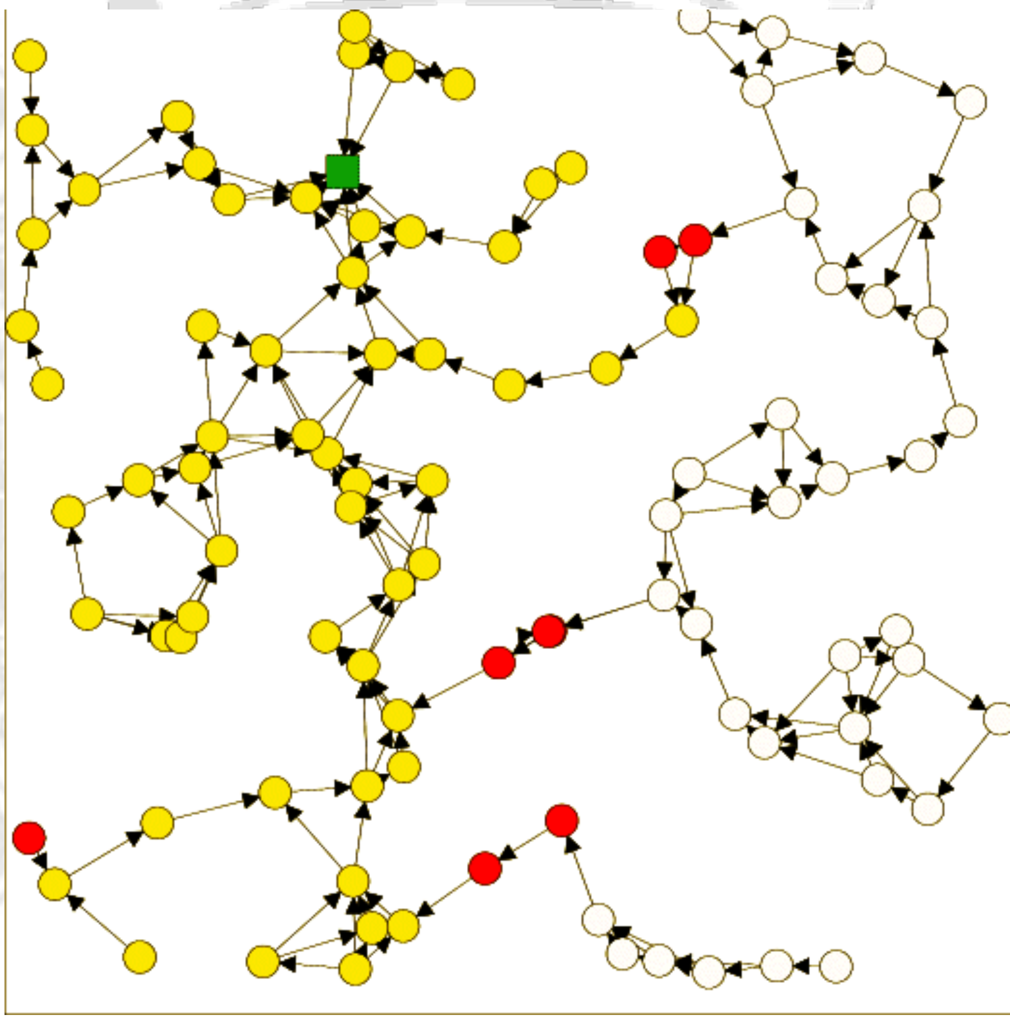




Rainbow Node Coloring Scheme: Red nodes

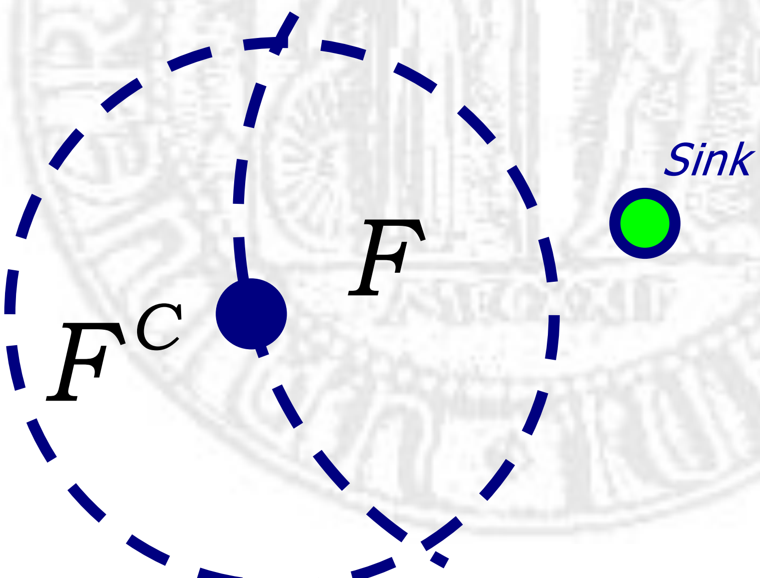
- If a yellow node cannot forward packets further, it switches to “red”
- From now, it looks for either “red” or “yellow” relays in F^C

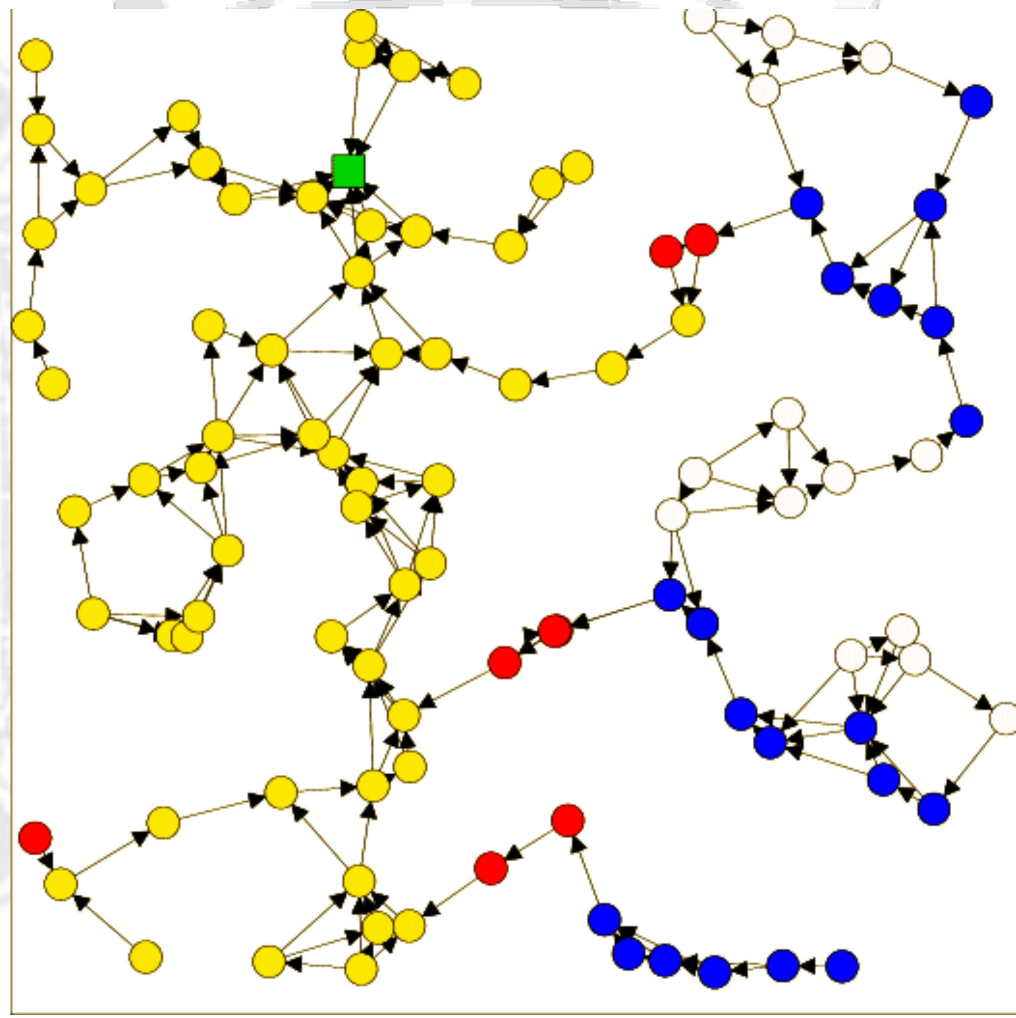




Rainbow Node Coloring Scheme: Blue Nodes

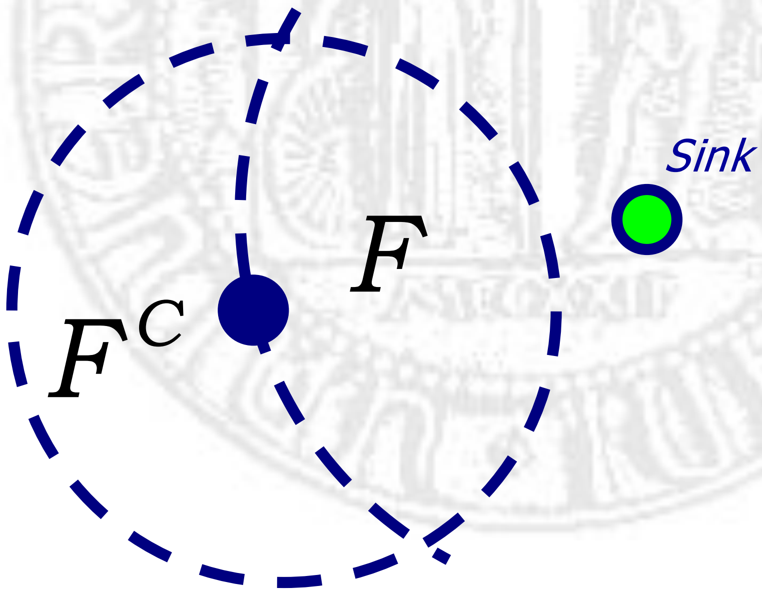
- If red nodes cannot advance packets, they turn to "blue"
- Again, they switch to look for relays in *F*
- They only look for "red" or "blue relays

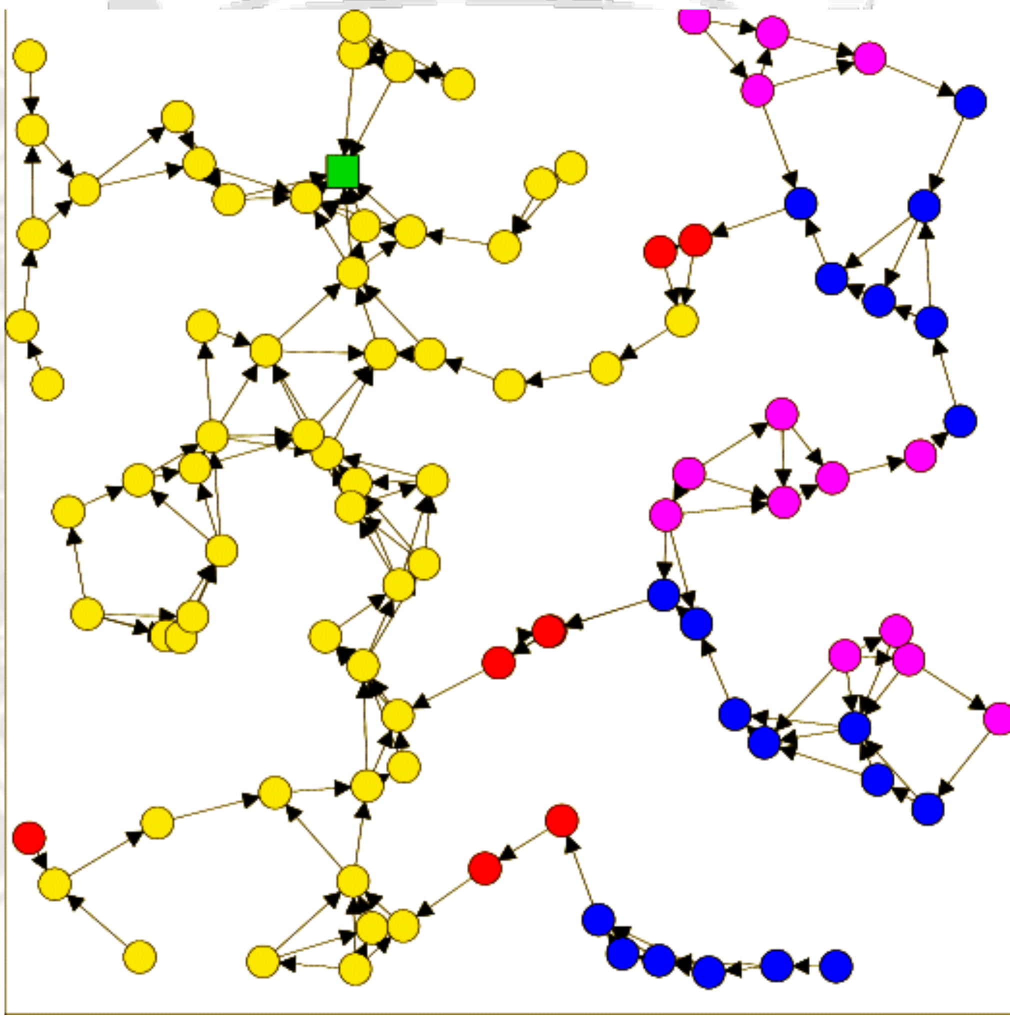




Rainbow Node Coloring Scheme: Violet nodes

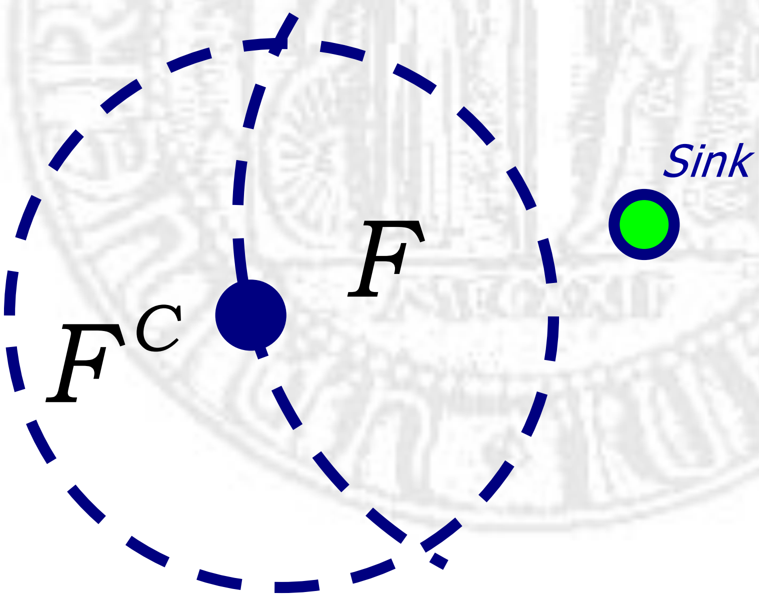
- If blue nodes still have problems finding relays they switch color again, to "violet"
- Like red nodes, they look for relays in F^C…
- …but only "blue" or "violet"





Rainbow Node Coloring Scheme: In general

- The number h of needed colors is fully general
  - The greater the number of colors, the more nodes can be connected to the converge-casting tree
- In general, given *h* labels $C_1, C_2, \ldots, C_h$ …
- The nodes switch from a label to the following one every time they perceive to be a dead end with their present label



- Nodes labeled C_1 are the only one with a greedy path to the sink
- Nodes with odd labels ($C_1, C_3, \ldots$) always look for relays in *F*
- Nodes with even labels ($C_2, C_4, \ldots$) always look for relays in *F*^{C}
- A node with label C_k always looks for C_k- or C_{k-1}-nodes, except C_1-nodes that always look for other C_1-nodes

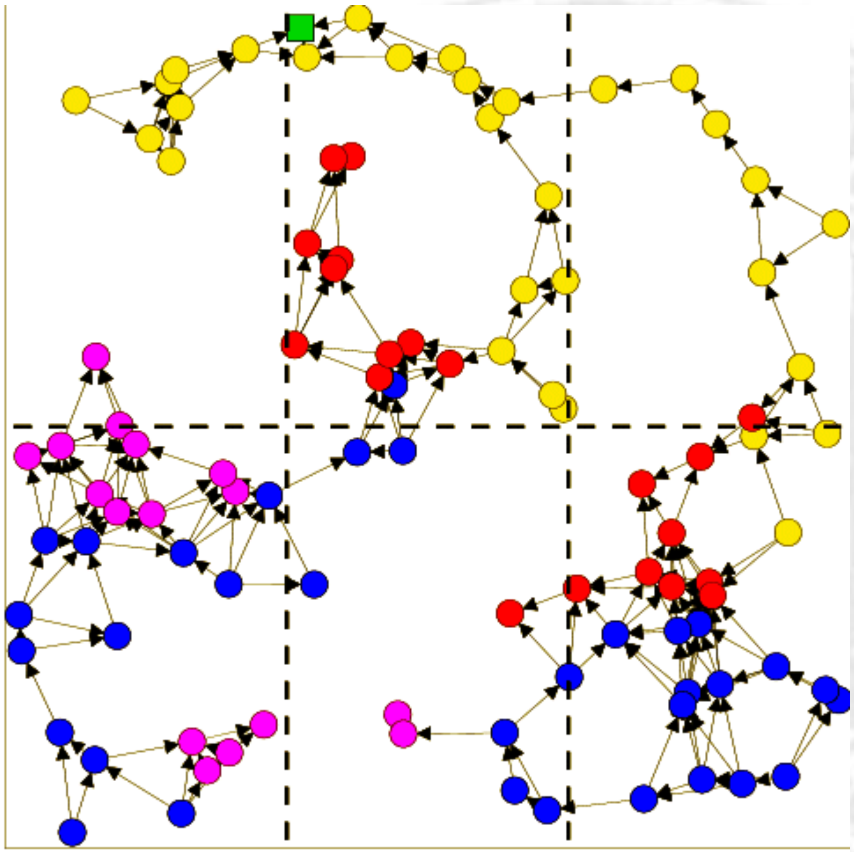
Rainbow: Wrap up

- **Concepts**
  - Nodes progressively realize to be dead end and automatically adapt to this condition
    - ✓ No abrupt changes in the color of a node (relays might be present but just *unavailable* for the moment)
  - More colors mean more nodes can successfully deliver packets
- **Pros**
  - Effectively routes around dead ends
  - Completely blind and distributed
  - Does not require planarization
  - The load-balancing features of ALBA are seamlessly used throughout
- **Cons**
  - The network requires some training for nodes to achieve the correct color

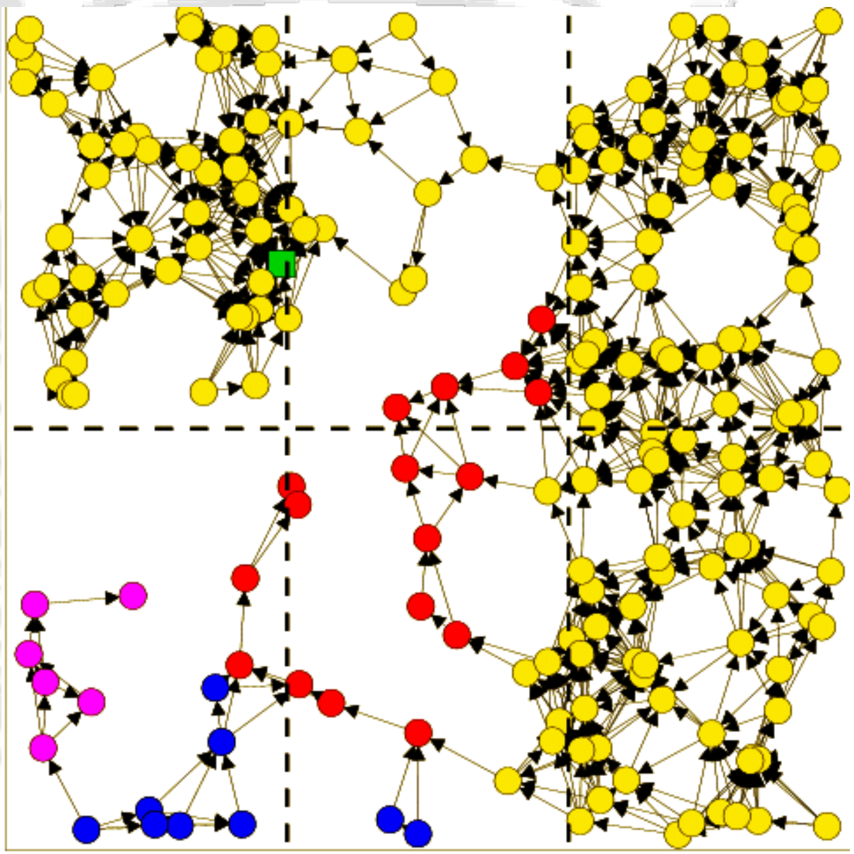
Results: Simulation Setting

- Simulation area: 320 m x 320 m
  - Random and uniform deployment
  - Non-uniform deployment
    - ✓ A more general case than uniform deployment
    - ✓ The area is divided in 3 high-density and 3 low-density zones
    - ✓ 75% of the nodes are randomly placed in high-density zones, the remaining 25% in low-density zones
- First set of results → Comparison
  - ALBA-R vs. GeRaF and MACRO
- Second set of results → High node densities
  - Show that Rainbow does not decrease performance if not used
  - N = 300, 600, 800, 1000 nodes
- Third set of results → Low node densities and different number of colors used in Rainbow
  - Used to show the effectiveness of Rainbow in rerouting packets

Sample non-Uniform Deployments



100 nodes



200 nodes