Localized Mobility Control Routing in Robotic Sensor and Actuator Wireless Networks

2007

Hai Liu
Amiya Nayak
Ivan Stojmenović
www.site.uottawa.ca/~ivan

Problem Specification

Fixed source and destination, long term traffic

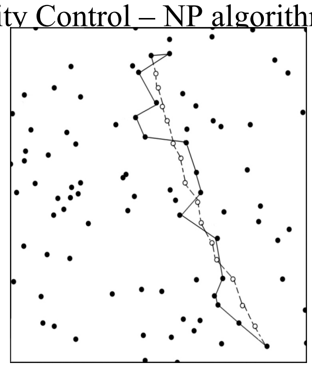
Mobile sensors, robots, actuators, human, vehicles … as intermediate nodes:

find a route and move each node on the route, such that

total transmission power is minimized,

total movement distance is minimized.

Routing Paths Before and After Mobility Control – NP algorithm



Assumptions

- common communication radius *r*.
- Energy cost model is $d^{\alpha} + c$ where *d* is distance
- Each node knows locations of its neighbors and its own location
- Energy to move is proportional to distance moved

Existing Solutions

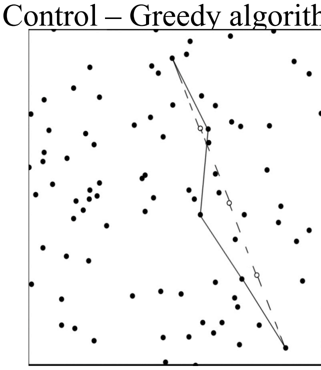
apply some routing (Greedy, NP) to establish an initial route; iteratively, each node (except for source and destination) moves to the midpoint of its upstream node and downstream node on the route.

Greedy (forward to neighbor closest to destination), or
NP (forward to nearest neighbors with progress)

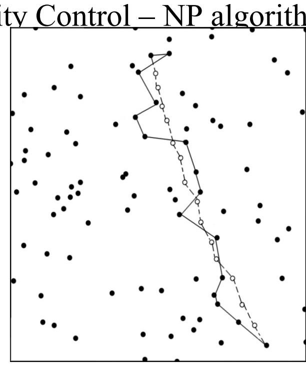
[GLM] D.K. Goldenberg, J. Lin, and A.S. Morse, "Towards Mobility as a Network Control Primitive," ***Mobihoc'04***, pp. 163-174, Japan, 2004.

Property: *connectivity preserved while moving*

Routing Paths Before and After Mobility Control – Greedy algorithm



Routing Paths Before and After Mobility Control – NP algorithm



Drawbacks and Motivation

- Initial route is not energy optimized
- Too many or too little forwarders
- Route after node movement may be far from energy optimal
- Iterative movement of nodes in rounds requires messages for synchronization and causes unnecessary zig-zig movement
- Large delay and possible communication failures

Contributions

- Study the optimal number of hops and optimal distance of adjacent nodes on the route.
- Propose OHCR algorithm which is based on the optimal **number of hops** on the route.
- Propose MPoPR algorithm which minimizes transmission **power** over progress.
- Study both strategies of **move in rounds** and **move directly**.

Overview of Our Solutions

two steps:

- compute **optimal number of hops** and **optimal distance of adjacent nodes** on the route.
- a routing algorithm that is based on the **optimal number of hops**, and
  a greedy algorithm that minimizes transmission **power over progress** in selecting a forwarding neighbor.

Optimal Number of Hops and Distance

Theorem 1. To minimize total transmission power of route from *s* to *t*, the optimal number of hops on the straight line route is integer *k*, minimizing

$$|k - d(s,t)\times((\alpha-1)/c)^{1/\alpha}|$$

and the optimal distance of adjacent nodes is $d(s,t)/k$, with energy cost model $d^\alpha + c$.

Optimal Hop Count Routing (OHCR)

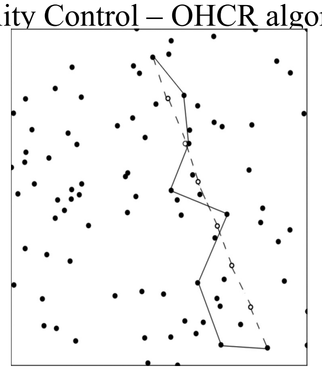
round $d(s,t)\times((\alpha-1)/c)^{1/\alpha}$ to the nearest integer *k*;

compute optimal distance of adjacent nodes $d(s,t)/k$;

if $k \le 0$ and $d(s,t) \le r$ *s* transmits directly to *t*;

current node *u* selects neighbor *v* such that $|d(u,v) - d(s,t)/k|$ is minimized;

Routing Paths Before and After Mobility Control – OHCR algorithm



Minimum Power over Progress Routing (MPoPR)

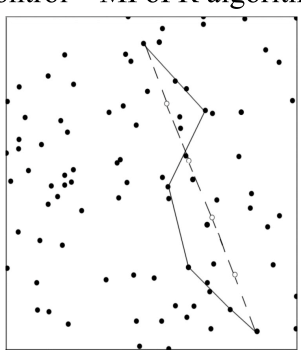
minimize transmission power of unit progress in selecting a forwarding neighbor.

u selects neighbor *v* such that

$$(d(u,v)^{\alpha} + c)/(d(u,t) - d(v,t))$$

is minimized;

Routing Paths Before and After Mobility Control – MPoPR algorithm



Move Directly Strategy

- Destination learns actual number of hops
- Which is routed backward to all nodes on route
- Nodes then learn actual target location to move
- Moves directly to decided position, no zig-zag
- However, the connectivity, while moving, is not guaranteed.

Conclusions and Future Work

- MPoPR is a good solution for move in rounds strategy while OHCR is good for move directly strategy.
- Move directly strategy costs less total energy than move in rounds strategy (but does not preserve connectivity during mobility).
- Use mobility control to improve network performance on other aspects, e.g., network capacity.
- Incorporate face routing into our algorithms to adapt sparse networks.