Fault-tolerant Relay Node Placement in Heterogeneous Wireless Sensor Networks

IEEE INFOCOM 2007

Xiaofeng Han, Xiang Cao, Errol L. Lloyd, Chien-Chung Shen

Presented By Chien-Hung Kuo

2007/10/24

Outline

- Introduction
- One-way PFRP
- Two-way PFRP
- One-way and two-way FFRP
- Extension to higher dimensions
- Performance
- Conclusion

Introduction

- The followings may cause sensors to fail
  - battery depletion
  - environmental impairment

- To achieve fault tolerance
  - Relay node placement

Introduction

- Most of the current work focus this placement on homogeneous WSNs.

  While this paper focus on Heterogeneous WSNs.
- There are two placements:
  - FFRP – Full Fault-tolerance Relay node Placement
  - PFRP – Partial Fault-tolerance Relay node Placement

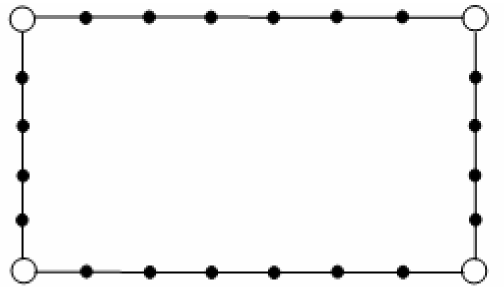
Introduction

- There are also two communication paths:
  - One way path
  - Two way path

- Each of these four problems is NP-hard, we develop some approximation algorithms for these problems

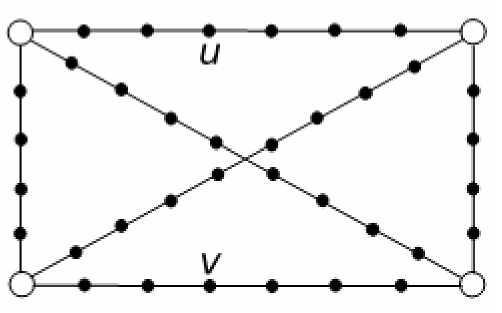
FFRP

- A full 2-vertex connected networks



PFRP

- A partial 3-vertex connected network



H-WSNs

- Heterogeneous WSNs
- Target nodes have different radii
- Relay nodes have the same radius
- Due to the above assumptions:
  - One-way paths
  - Two-way paths

One-way PFRP

- At first, create a complete graph
- Using the following equation to define the weight of each edge

Equation 1.

$$weight(\overrightarrow{uv}) = \begin{cases} 0 & \text{if } T(u) \geq |uv| \\ \lceil \frac{|uv| - T(u)}{T(relay)} \rceil & \text{if } T(u) < |uv| \end{cases} \quad (1)$$

One-way PFRP

- Let C be a complete graph containing the above weighted edges and target nodes
- Compute an approximation directed MKCSG M of C using a p-approximation algorithm
- One-way steinerize each edge and place the relay nodes
- This is an $O(pk^2)$-approximation algorithm

MKCSG

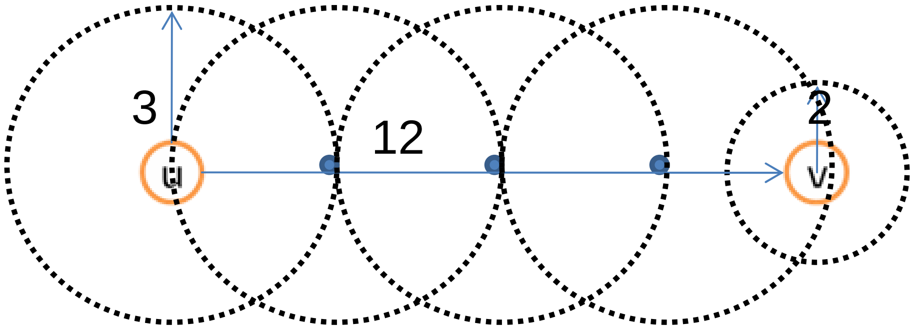
- Using greedy algo
- At first, add all edges in C with weight zero
- Repeatedly add the edge with highest contribution
- Test each edge in decreasing order of weight

Contribution

- The contribution of an edge $\overrightarrow{uv}$ is defined as the number of unsaturated node pairs.

- Unsaturated node pairs:
  - The connectivity is lower than k

One-way steinerization



Two-way PFRP

- Similar to One-way PFRP
- Using equation 2 to define the weight

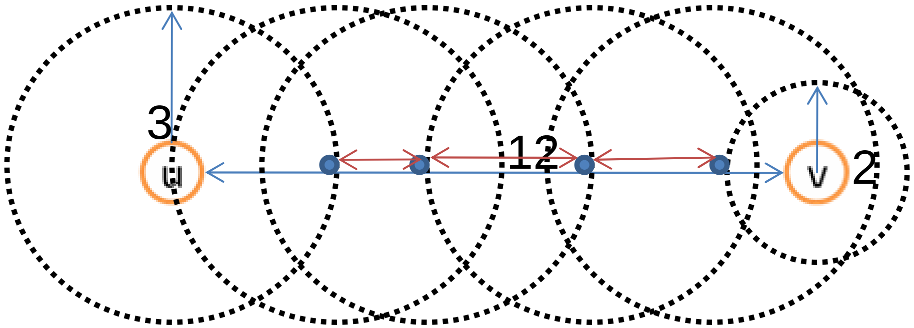
$$\delta = min\{T(u), T(v)\},$$

$$\lambda = min\{T(u), T(relay)\} \text{ and } \omega = min\{T(v), T(relay)\}.$$

$$weight(\widehat{uv}) = \begin{cases} 0 & \text{if } |uv| \leq \delta \\ \lceil \frac{|uv| - \lambda - \omega}{T(relay)} \rceil + 1 & \text{if } |uv| > \delta \end{cases} \quad (2)$$

- Then, do two-way steinerization

Two-way steinerization



One-way (Two-way) FFRP

- Partial:
  - one target node to another target node
- Full: additional viewpoints
  - one target node to another relay node
  - one relay node to another relay node

One-way (Two-way) FFRP

- Execute One-way PFRP algo on V and get a set of relay nodes R

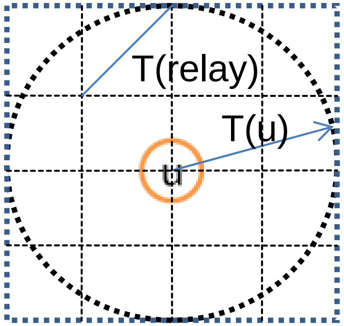
- Place k-1 additional relay nodes at the position of every relay node in each super path

One-way (Two-way) FFRP

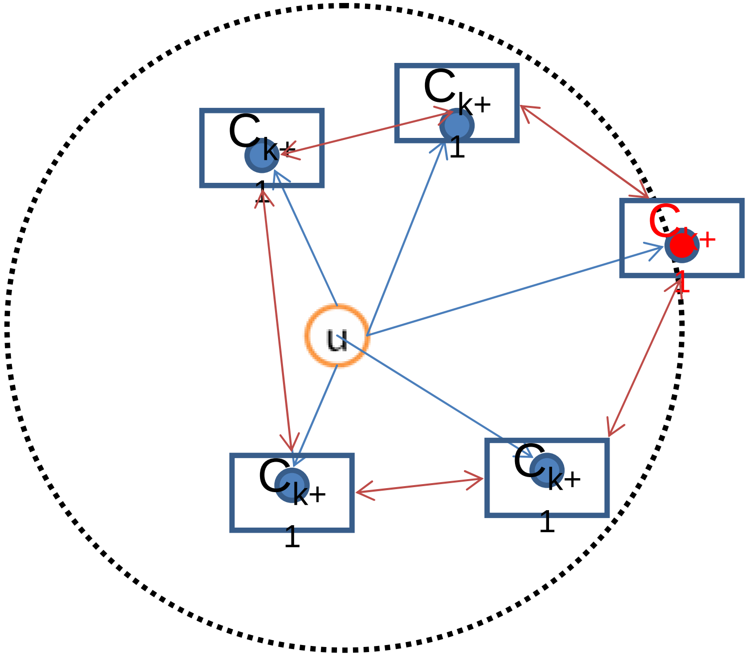
- For each starting or ending node u of a super path, we segment u’s neighborhood with (T(u),T(relay))
- Place a cluster of k-1 relay nodes at the position of u
- Place a cluster of k relay nodes at the center of each cell

Segmentation

- Node u is a target node
- Nodes in each cell are all relay nodes besides node u.



Segmentation



One-way (Two-way) FFRP

- Finally, for each cluster
  - Check if the resulting network is k-vertex connected when remove all relay nodes in the cluster
  - If so, remove them

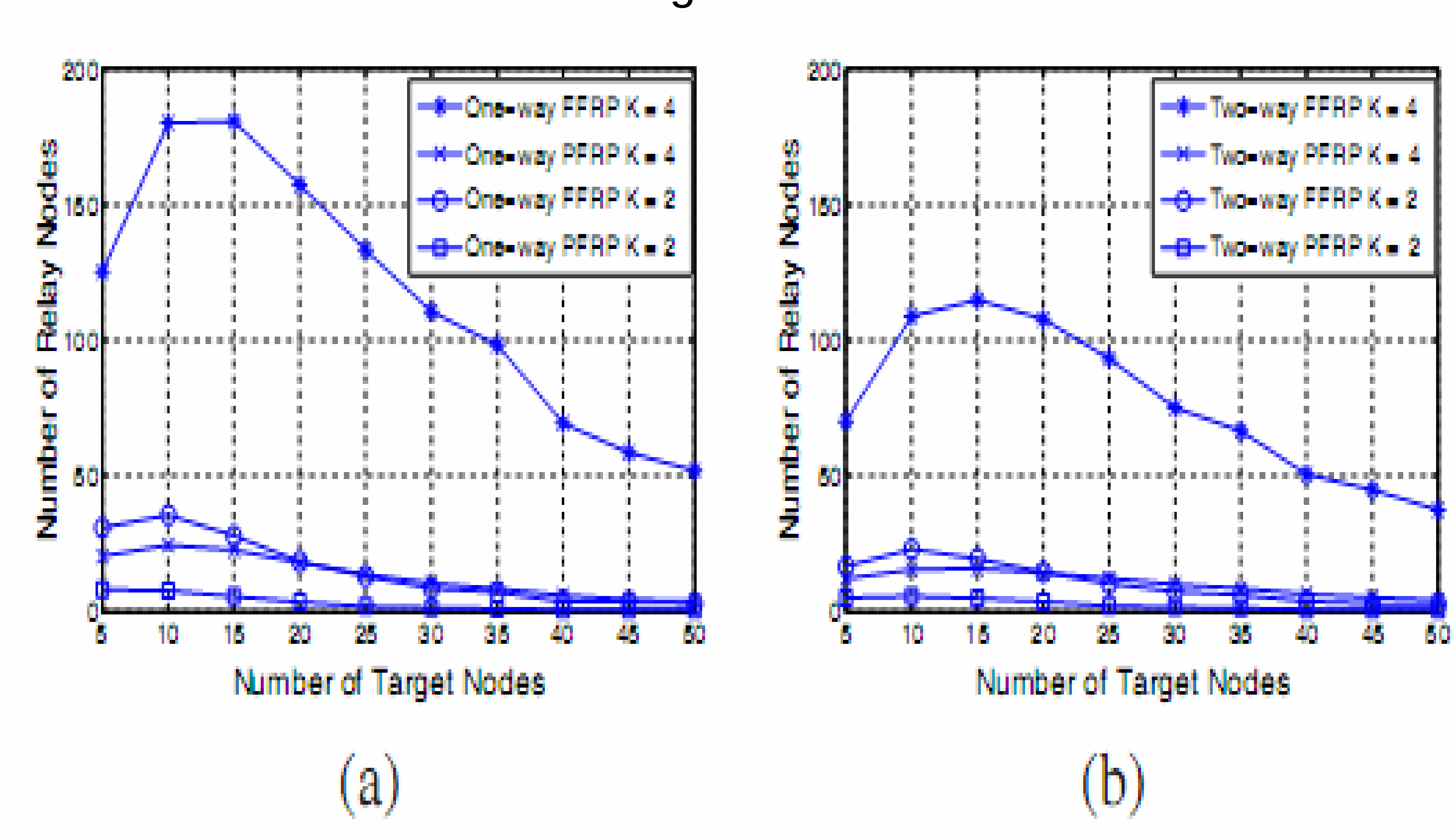
- This is an $O(pk^3)$-approximation algorithm

Performance

- Use Qualnet 3.8 as the simulation platform
- Randomly place target nodes in a 1000m x 1000m 2D terrain
- $T(min) = 200m$
- $T(max) = 500m$
- $T(min) <= T(u) <= T(max)$
- 50 runs

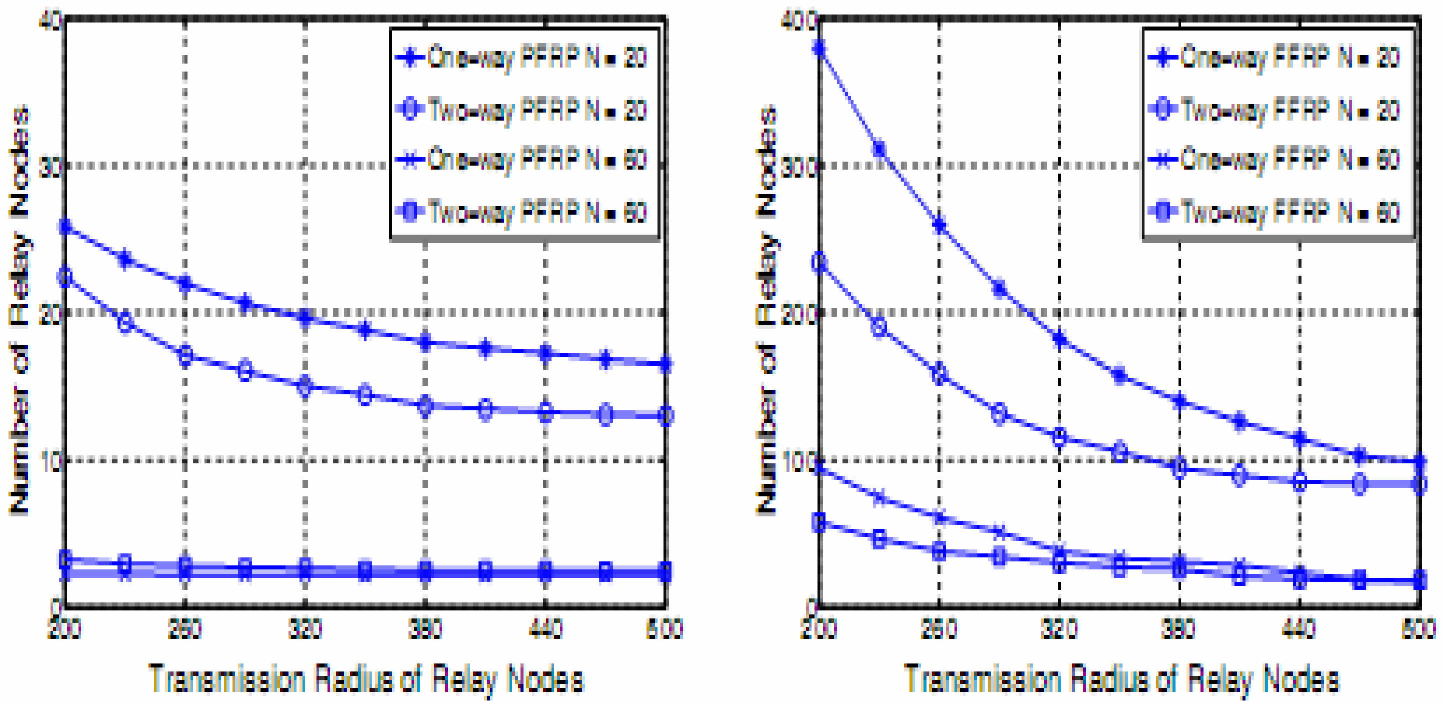
Performance

T(relay) = 350m
The number of target nodes is from 5 to 50



Performance

T(min)≤T(relay) ≤T(max)
The number of target nodes is 20 and 60



Conclusions

- Pursue tighter performance ratios of the approximation algo

- Pursue better heuristic implementation