Connected Dominating Sets in Wireless Networks

My T. Thai
Dept of Comp & Info Sci & Engineering
University of Florida

June 20, 2006
Wireless Ad Hoc Networks

- A collection of mobile nodes
- Dynamically form a temporary network
Wireless Sensor Networks (WSN)

- Consists of a large number of sensor nodes
- Main Tasks: collaborate to sense, collect, and process the raw data of the phenomenon and transmit the processed data to sinks
Applications

- Military applications
- Environmental applications
- Health applications
- Other commercial applications
Characteristics of Wireless Ad Hoc Networks

- Dynamic topology – no predefined or fixed infrastructure
- Multi-hop routing – each node is a router
- Limited resources – battery power, CPU, storage, and bandwidth

Routing decision is challenging!
Virtual Backbone
Virtual Backbone Features

- **Minimize** the virtual backbone nodes
- All virtual backbone nodes are **connected**
- Each node is either in or adjacent to the backbone
- Approximated by **Minimum Connected Dominating Set (MCDS)**
Definitions

- Given a graph $G=(V,E)$ and a subset $C \subseteq V$. $C$ is:
  - **Dominating Set (DS):** for any $v \in V$, $v \in C$ or adjacent to some $u \in C$
  - **Connected Dominating Set (CDS):** $C$ is a DS and an induced graph of $C$ is connected
  - **Minimum Connected Dominating Set (MCDS):** $C$ is a CDS and has the smallest size
Approximation Algorithms

- An algorithm that returns near-optimal solutions in polynomial time

- **Performance Ratio (PR):**
  - Minimization problem: $|C|/|C^*|$ where:
    - $C$ is a near-optimal MCDS
    - $C^*$ is the optimal MCDS
  - Smaller PR, better algorithm
Homogeneous Networks

What if all nodes have the same transmission ranges?

Can we design a constant approximation algorithm?
Unit Disk Graphs (UDG)

- **UDG**: is an intersection graphs of circles of unit radius in the plane

- **Lemma 1**: Each node in a UDG has at most 5 independent neighbors
Maximal Independent Set (MIS) is a maximal set of pairwise non-adjacent nodes.

\[\text{MIS} \rightarrow \text{DS}\]
Algorithm 1 - Overview

- **Phase 1**: Construct an MIS such that:
  - **Lemma 2**: Any pair of complementary subsets of the MIS separate by exactly two hops
- **Phase 2**: Connect MIS $\rightarrow$ CDS
Algorithm 1 – Phase 2

- **Goal:** Connect an MIS by adding the minimum number of blue nodes where:
  - Blue Nodes: Nodes connecting black nodes
- **Black-blue component:** a connected component of the sub-graph induced only by black and blue nodes
Algorithm 1 – Phase 2 (cont)

Input: An MIS:
- All nodes in MIS are black
- Others are grey

for i=5, 4, 3, 2 do
  while there exists a grey node adjacent to at least $i$ black-blue components do
    change its color from grey to blue
    re-construct the black-blue components
  return all black and blue nodes
Algorithm 1 – An Example
Algorithm 1 - Analysis

- **Theorem 1:** Algorithm 1 has a performance ratio of $5.8 + \ln 4 < 8$
  - Lemma 3: $|\text{MIS}| \leq 3.8|C^*| + 1.2$
  - Lemma 4: $\#\text{Blue Nodes} \leq (2 + \ln 4)|C^*|$
Fault Tolerance

What if a virtual backbone node is dead?

What if a link in the virtual backbone is broken?
2-CDS

Problem Definition:

- Given a UDG $G=(V,E)$
- Find a CDS $C$ satisfying:
  - $|C|$ is minimum
  - For any pair of nodes in $C$, there exists 2-disjoint paths
Algorithm 2 - Overview

- **Phase 1**: Construct a CDS $C$
- **Phase 2**: Augment $C$ to obtain a 2-CDS
Algorithm 2 – Phase 2

- **Cut-vertex**: $x$ is a cut-vertex if $G$-$\{x\}$ is disconnected
- **Block**: a maximal subgraph of $G$ without cut-vertices
- **Leaf Block**: a maximal subgraph of $G$ with exactly 1 cut-vertex
Algorithm 2 – Phase 2 (cont)

while C has more than 1 blocks do
    \( L = \) Leaf Block
    Find a shortest path \( P = ux_i\nu \) where \( v \in L \), \( v \) is not a cut-vertex, \( u \in C \setminus L \), and \( x_i \in V \setminus C \)
    color \( x_i \) blue
end while

return all black and blue nodes
Theoretical Analysis

- **Theorem 2:** Algorithm 2 has a constant performance ratio of 62
Simulation Experiments

- Randomly deployed nodes into 1000 x 1000 m² region
- Transmission range = 200 m
- Each setting, ran 1000 times

Algorithm 2 improves the fault tolerance of virtual backbone with only marginal extra overhead
More Challenging

What if networks have unidirectional links and different transmission ranges?

Can we design a constant approximation algorithm?
Heterogeneous Networks

- Model networks as **Disk Graphs**:
  - Each node has a transmission range in \([r_{\text{min}}, r_{\text{max}}]\)
  - A directed edge from \(u\) to \(v\) iff \(d(u,v) \leq r_u\)

- Bidirectional Links and Unidirectional Links
Unidirectional Links

- Directed graph $G = (V, E)$
- **Strongly Connected Dominating Set (SCDS):**
  - Given a directed graph $G = (V, E)$
  - Find a subset $C \subseteq V$ such that:
    - $\forall v \in V, v \in C$ or there exists a node $u \in C$ such that $uv \in E$
    - The subgraph induced by $C$ is strongly connected, i.e., there exists a directed path for any pair of nodes in $C$
SCDS – An Example
Greedy Algorithm 3 - Overview

- **Phase 1**: Construct a DS $D$
- **Phase 2**: Connects all nodes in $D$ to form a SCDS $C$
Algorithm 3 – Phase 1

while there exists a white node do

select a white node \( u \) with the biggest transmission range

color \( u \) black

color all \( N^+(u) \) grey

end while

return all black nodes
Algorithm 3 – Phase 2

Goal: Connect a DS $D$ by adding the minimum number of blue nodes

- Let $u \in D$ s.t. $u$ has the largest transmission range
- Build a Minimum nodes Directed Tree (MDT) $T_1$ rooted at $u$ s.t. there is a directed path from $u$ to all other nodes in $D$
- Construct $G'$ from $G$ by reversing the directed edges
- Build a MDT $T_2$ rooted at $u$
- All nodes in the union of $T_1$ and $T_2$ form a SCDS $C$
Minimum nodes Directed Tree (MDT)

- Given a directed graph \( G = (V, E) \), a subset \( D \) of \( V \), and a node \( u \)
- Find a tree \( T \) rooted at \( u \) such that:
  - There exists a directed path from \( u \) to all nodes in \( D \)
  - The total number of nodes in \( T \) but not in \( D \), called blue nodes, is minimum
An MDT Algorithm

- Denote:
  - All nodes in $D$ are black
  - All nodes in $T \setminus D$ are blue
  - Other nodes are grey

- $v$-spider: A directed tree having:
  - one grey node $v$ as a root
  - other nodes are either black or blue
  - there is a directed path from $v$ to all other nodes in the spider
An MDT Algorithm (cont)

- Contracting Operation:
  - Add a directed edge from \( v \) to all grey nodes that are outgoing neighbors of blue and black nodes in \( v \)-spider
  - Delete all black and blue nodes and their edges
  - Color \( v \) blue
An MDT Algorithm - Description

while no directed paths from $u$ to $D$ in $T$ do

    Find a $v$-spider that has the most number of blue and black nodes

    Contract this $v$-spider

    Construct $T$ from the set of black and blue nodes

end while
Algorithm 3 - Analysis

- **Theorem 3**: The size of any DS is upper bounded by:
  \[ 2.4 \left( k + \frac{1}{2} \right)^2 |C^*| + 3.7 \left( k + \frac{1}{2} \right)^2 \]
  where \( k = \frac{r_{\text{max}}}{r_{\text{min}}} \)

- **Theorem 4**: Algorithm 4 has a performance ratio of
  \[ 2.4 \left( k + \frac{1}{2} \right)^2 + 4 + 4 \ln(2k - 1) \]
More Work

- k-connected m-dominating set
  - In the presentation: 2-connected 1-dominating set

- k-connected m-dominating set in heterogeneous networks
Thank You!

Any Questions?
Benefits of Virtual Backbone

**Broadcast**

- Only a subset of nodes (virtual backbone nodes) relay messages:
  - Reduce communication cost
  - Reduce redundant traffic
  - Conserve energy
Benefits of Virtual Backbone (cont)

Unicast

- Only a subset of nodes maintain routing tables
- Routing information localized
  - Save storage space