Constructing $K$-Connected $M$-Dominating Sets in Wireless Sensor Networks

Yiwei Wu, Feng Wang, My T. Thai and Yingshu Li

Georgia State University
Arizona State University
University of Florida
• Introduction
• Centralized Algorithm – \textit{CGA}
• Distributed Algorithm – \textit{DDA}
• Simulation Results
• Conclusion
Introduction

• A Connected Dominating Set (CDS) $C$ of $G$ is a dominating set of $G$ which induces a connected subgraph of $G$. The nodes in $C$ are called dominators, the others are called dominatees.

• A CDS is the earliest proposed candidate to serve as a virtual backbones in WSNs.
Using this virtual backbone, a sender can send messages to its neighbouring dominator. Then along the CDS, the messages are sent to the dominator closest to the receiver. Finally, the messages are delivered to the receiver.

$k$-Connected $m$-Dominating Set ($kmCDS$) is necessary for fault tolerance and routing flexibility.
Introduction

• \(k\)-connected or \(k\)-vertex connected
  – A graph is \(k\)-connected if and only if it contains \(k\) independent paths between any two vertices.

• \(m\)-dominating set
  – If each node not in \(C\) is dominated by at least \(m\) nodes in \(C\), then \(C\) is a \(m\)-dominating set.

• \(k\)-connected \(m\)-dominating set
Outline

• Introduction
• **Centralized Algorithm – CGA**
• **Distributed Algorithm – DDA**
• Simulation Results
• Conclusion
Centralized Algorithm CGA

• **CGA** is a greedy algorithm with time complexity of **CGA** is $O(|V|^{3.5}|E|)$.

• The main idea:
  - construct an $m$-dominating set $C$
  - augment this set $C$ for $k$-connectivity
  - remove redundant nodes (optional)
Centralized Algorithm CGA

- **Notations:**
  - $N_i$: The number of neighbors of a node $i$.
  - $e_i$: The energy of a node $i$.
  - $N^c_i$: The number of dominator neighbors of node $i$.
  - $ID_i$: The ID of node $i$.
  - $C$: A $k$-connected $m$-dominating set.

- **Weight function** $w(N, e, ID)$
Algorithm 1 CGA\((k, m, G(V, E))\)

1: \textbf{procedure} FINDKMCDS\((k, m, G(V, E))\)
2: \hspace*{1em} Sort nodes in non-increasing order in \(G\) based on their \((N_i, e_i, ID_i)\)
3: \hspace*{1em} \(C \leftarrow \emptyset\)
4: \hspace*{1em} \textbf{for} \(i = 1\) to \(|V|\) \textbf{do}
5: \hspace*{2em} \textbf{if} \(N_i^c < m\) \textbf{then}
6: \hspace*{3em} \(C \leftarrow C \cup \{v_i\}\)
7: \hspace*{2em} \textbf{end if}
8: \hspace*{1em} \textbf{end for}
9: \hspace*{1em} \textbf{while} \(C\) is not \(k\) connected \textbf{do}
10: \hspace*{2em} \textbf{add a dominatee with highest} \((N_i, e_i, ID_i)\) \textbf{into} \(C\)
11: \hspace*{1em} \textbf{end while}
12: \hspace*{1em} \textbf{return} \(C\)
13: \textbf{end procedure}
Outline

• Introduction
• Centralized Algorithm – CGA
• Distributed Algorithm – DDA
• Simulation Results
• Conclusion
Distributed Algorithm - DDA

• *DDA* consists of three phases
  – Phase 1: Use one of the distributed *CDS* algorithms to construct a *CDS C*.
  – Phase 2: Augment *C* to a 1-connected *m*-dominating set by adding *m*-1 *MISs*.
  – Phase 3: Connect set *C* for *k*-connectivity.
• How to make $C$ for $k$-connectivity in a distributed way?

• Lemma 1: If $G$ is a $k$-connected graph, and $G'$ is obtained from $G$ by adding a new node $v$ with at least $k$ neighbors in $G$, then $G'$ is also a $k$-connected graph.
Distributed Algorithm – DDA phase 3

• Main idea:
  – The leader builds a $k$-connected component.
  – In order to join the $k$-connected component, one black node adds at most $k^2$ connectors according to the previous lemma.
Distributed Algorithm – DDA phase 3

• Important messages used in DDA:
  – K-ConnectedComponent (KC) message
  – RequireConnector (RC) message
  – ACKConnected (AC) message
  – ConfirmSuccess (CS) message
  – ConfirmUnuse (CU) message
Distributed Algorithm – DDA phase 3

- State transition diagram for black nodes
Distributed Algorithm – DDA phase 3

• State transition diagram for white nodes
Distributed Algorithm – DDA

• An example \((k=2)\)
Distributed Algorithm – DDA

- An example ($k=2$)

![Diagram showing a distributed algorithm example with nodes labeled 1 to 6, CS at node 3, ACK at node 4, and a K-connected component highlighted.]
Distributed Algorithm – DDA

• An example \((k=2)\)
• **Lemma 2**: Every subset of an *MIS* is at most three hops away from its complement.

![Diagram showing 5 nodes connected in a line, with some nodes shaded.]  

• **Lemma 3**: Let $G = (V, E)$ be any *UDG* and $m$ be any constant such that $\delta_G \geq m - 1$ where $\delta_G$ is the minimum node degree of graph $G$. Let $D^*_m$ be any optimal $m$-domination of $G$ and $S$ be any MIS of $G$. Then $|S| \leq 5m|D^*_m|$. 
Distributed Algorithm – DDA

• **Theorem**: If $C$ is a $kmCDS$ obtained by DDA, then $|C| \leq 5m(k^2 + 1)(m+42)opt$, where $opt$ is the size of any optimal $kmCDS$ of the network.

• **Proof:**
  - Phase 1: $43|S|$
  - Phase 2: $(m - 1)|S|$
  - Phase 3: $k^2$ for each black node in phase 1 and 2
• The message complexity of \textit{DDA} is $O(|V|\Delta^2)$ and time complexity is $O(m\Delta + \text{Diam})$, where $\Delta$ is the maximum node degree and $\text{Diam}$ is the diameter of the network.
Outline

• Introduction
• Centralized Algorithm – *CGA*
• Distributed Algorithm – *DDA*
• Simulation Results
• Conclusion
Simulation and Results

- $1000m \times 1000m$ area; transmission range $= 250m$
Simulation and Results

- $1000m \times 1000m$, transmission range $= 350m$
Simulation and Results

- Compare *CGA* with *CDSA* when $k = 2, m = 1$
Simulation and Results

- Comparision of DDA with $k$-Coverage ($k=2$).

![Graph showing comparison between DDA and $k$-Coverage.](image)
Simulation and Results

- Comparision of *DDA* with *k-Coverage* (*k=3*).
Outline

• Introduction
• Centralized Algorithm – CGA
• Distributed Algorithm – DDA
• Simulation Results
• Conclusion
Conclusion

• We investigate the problem of constructing a $km$CDS for general $k$ and $m$.

• We propose one centralized algorithm $CGA$ and one distributed algorithm $DDA$.

• Our algorithms can obtain good results.
Questions?

Thanks!