A DOMINATION SET APPROACH TO DATA AGGREGATION IN NETWORKED EMBEDDED SYSTEMS

A Thesis
Presented to
The Graduate Faculty of The University of Akron

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

Maithili Ghamande
May, 2010
ABSTRACT

Networked embedded systems comprise a large number of deeply embedded nodes that are interconnected via low power, low bandwidth, wireless links. Data from these nodes must be collected for monitoring and decision-making in applications such as real-time surveillance. It is not feasible for every node to directly send all its data to one or more collection points because of the data volume and power constraints at each node, and the low bandwidth of the wireless links. Data aggregation techniques enable the gathering and processing of data in such contexts.

The use of cluster heads (CHs) for data aggregation is an important approach that is widely reported in the literature. A cluster head is a special node in the system that can aggregate data in its neighborhood and forward the aggregated data towards the collection points. Depending on how the CHs are connected, the network of CHs can either be flat or hierarchical. Nevertheless, the CHs are selected either in an ad-hoc manner, or by using an estimate of the remaining energy in the nodes. The current approaches do not exploit the topology of the system to select CHs. The idea of dominating sets in graphs offers an alternative, systematic, approach to identifying CHs.
This thesis considered regular mesh topologies in which each node has \( q \) neighbors, \( q = 3, 4, 6, 8 \). For each of these topologies, intuitive patterns which enable an easy determination of the dominating sets and the location of the CHs, are presented. It is shown that the parameter \( k \) impacts the energy efficiency and the Quality of Service (QoS) of the data aggregation scheme. Simulation results provide insights into the effects of the topology. In the future, this work can be extended to address data aggregation in general topologies.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF TABLES</td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Contributions</td>
<td>4</td>
</tr>
<tr>
<td>1.2 Overview of the Thesis</td>
<td>4</td>
</tr>
<tr>
<td>II. BACKGROUND</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Regular Mesh Topologies</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Related Work</td>
<td>7</td>
</tr>
<tr>
<td>III. SYSTEM MODEL AND METRICS</td>
<td>12</td>
</tr>
<tr>
<td>3.1 Nodes and Collection Points</td>
<td>12</td>
</tr>
<tr>
<td>3.2 Data Aggregation Scheme</td>
<td>13</td>
</tr>
<tr>
<td>3.3 Energy Metrics</td>
<td>14</td>
</tr>
<tr>
<td>IV. IDENTIFYING CLUSTER HEADS IN REGULAR MESH TOPOLOGIES</td>
<td>16</td>
</tr>
<tr>
<td>4.1 Definitions and Approach</td>
<td>16</td>
</tr>
<tr>
<td>4.2 Minimal $k$-Dominating Sets</td>
<td>17</td>
</tr>
</tbody>
</table>
4.3 Viability of Clusters ........................................... 22

V. RESULTS ...................................................... 25

5.1 QoS Metrics ................................................. 25

5.2 Simulation Approach ........................................ 26

5.3 Effect of $k$ on QoS ........................................ 27

VI. CONCLUSION ............................................... 41

BIBLIOGRAPHY .................................................. 43
LIST OF TABLES

Table | Page |
--- | --- |
2.1 Embedding functions that embed a collection of nodes $n_i \in \mathcal{N}$ on a 2D-Basegrid with $R$ rows and $C$ columns [12]. | 7 |
2.2 Comparing Hierarchical and Flat networks. | 8 |
4.1 In mesh topologies $\Xi_3$, $\Xi_4$, $\Xi_6$, and $\Xi_8$, select nodes $n_{X,Y}$ as cluster heads as the initial set of nodes for the set $D$. | 20 |
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Regular Mesh Topologies obtained using embedding functions $\Xi_4$ and $\Xi_3$ [12]</td>
</tr>
<tr>
<td>4.1</td>
<td>The diamonds represent the tiles and, hence, $\text{Cover}(C)$ of $N$ in $\Xi_4$. The $k$-semitiles at the boundaries are shown with dotted outlines.</td>
</tr>
<tr>
<td>4.2</td>
<td>Cluster Head selection in $\Xi_8$ (left) and $\Xi_6$ (right) based on 1-domination.</td>
</tr>
<tr>
<td>4.3</td>
<td>Tiling pattern for 1-domination.</td>
</tr>
<tr>
<td>4.4</td>
<td>Shaded vertices as cluster heads for vertices not 2-dominated in $Z$ using shortest path algorithm.</td>
</tr>
<tr>
<td>5.1</td>
<td>Latency in $\Xi_4$ with $k=2$ on a 20x20 grid.</td>
</tr>
<tr>
<td>5.2</td>
<td>Latency in $\Xi_4$ with $k=5$ on a 20x20 grid.</td>
</tr>
<tr>
<td>5.3</td>
<td>Jitter in $\Xi_6$ with $k=1$ on a 20x20 grid.</td>
</tr>
<tr>
<td>5.4</td>
<td>Jitter in $\Xi_6$ with $k=5$ on a 20x20 grid.</td>
</tr>
<tr>
<td>5.5</td>
<td>Maximum throughput achieved in $\Xi_8$ for when $k=1$ in 20x20 grid.</td>
</tr>
<tr>
<td>5.6</td>
<td>Throughput decreases when $k=1$ in $\Xi_8$.</td>
</tr>
<tr>
<td>5.7</td>
<td>Percentage message loss in $\Xi_3$ for $k=2$ on a 20x20 grid.</td>
</tr>
<tr>
<td>5.8</td>
<td>Percentage message loss in $\Xi_3$ for $k=5$ on a 20x20 grid.</td>
</tr>
<tr>
<td>5.9</td>
<td>Latency comparison between $\Xi_3$, $\Xi_4$, $\Xi_6$, and $\Xi_8$ for $k=1$.</td>
</tr>
</tbody>
</table>
5.10 Latency comparison between $\Xi_3$, $\Xi_4$, $\Xi_6$, and $\Xi_8$ for $k=3$. ............... 35

5.11 Percentage message loss in $\Xi_4$. .............................................. 35

5.12 Percentage message loss in $\Xi_3$. .............................................. 36

5.13 Percentage message loss with $k = 2$ on a 10x10 grid. ...................... 36

5.14 Percentage message loss with $k = 2$ on a 20x20 grid. ...................... 37

5.15 Average Latency and Jitter for $\Xi_4$, minimum latencies and jitter are observed when $k = 1$, because of the single hop network around the cluster head compared to the dense networks at higher $k$’s. The outlier performances observed is because of the semi-tiles in the grid. .... 37

5.16 Average throughput for $\Xi_8$, the throughput decreases as the number of nodes in the cluster increase. ................................. 38

5.17 Throughput comparison with $k=2$ and $k=5$. Higher throughputs are achieved by topologies which contribute more messages to the cluster heads in a short time like $\Xi_4$ and $\Xi_8$. ......................... 38

5.18 Jitter at Sink level for $\Xi_3$ ............................................................... 39

5.19 Latency at Sink level for $\Xi_3$ and $\Xi_6$ ............................................ 39

5.20 Throughput at Sink level for $\Xi_4$ ................................................... 40
CHAPTER I

INTRODUCTION

Networked Embedded Systems have wide applications in a broad spectrum of areas including military, surveillance, medical, environmental systems. The nodes in such systems are distributed spatially and temporally and are interconnected over low power, low bandwidth, wireless links. The computation, communication, and power resources in many of the nodes may be severely constrained. Collecting data from all the nodes in the system and efficiently integrating this data for decision-making is an important function for many applications.

Because of the limited resources in each node, and the low bandwidth, it is not feasible for every node to send all its data to one or more collection points. The low power constraint requires that data from the nodes must travel over multihop routes towards the collection points. If many of these routes pass through the same node, such common nodes become critical and tend to become disabled because of their low energy constraints. Consequently, the network lifetime is severely impaired because of the loss of connectivity. Further, because the network bandwidth is low, in-network processing techniques are necessary to reduce the data traffic that moves through the links. While on the one hand, it is important to maintain high fidelity by not loosing data from the nodes, it is simultaneously necessary, on the other hand,
to extend the network lifetime while maintaining a high Quality of Service (QoS). Data aggregation techniques effectively process the network data, improve energy efficiency, and deliver good QoS.

One of the important considerations in data aggregation is to determine the nodes on which the aggregation techniques must be executed. In the literature [13], these nodes are typically referred to as Cluster Heads (CHs). Each CH in the system serves as a local collection point in its neighborhood. If the CH nodes can be designed to be more resourceful than other nodes, then, data from all the nodes can be sent to a collection point using a two-tier scheme. In the first tier, CHs gather their data from local neighborhood, and in the second tier, CHs send the gathered, and processed, data to the collection point. This simple idea of using CHs and a two-tier approach to data collection raised many interesting issues related to accuracy and freshness of the data, QoS, security, reliability and fault tolerance. All these aspects have been reported in the literature. A key issue in these approaches is - how to select the CHs? Many approaches reported in the literature rely either on ad-hoc techniques or the estimated remaining energy in the node as a basis for the selection. These approaches do not exploit the topology, i.e., the structure of the interconnection among the nodes, of the system. This investigation focused on an alternative, systematic, approach that is based on the idea of dominating sets in graphs.

By viewing the topology of the system as a graph, the approach presented selects CHs using the $k$-dominating sets of the graph. Let $G = (V, E)$ represent a graph in which $V$ represents the set of nodes. Each link between a pair of nodes is
represented as an edge \( e \in E \). A subset \( D \subset V \) is called a 1-dominating set if every node \( v \in V \) is either in \( D \), or is adjacent to a node in \( D \), i.e., \( (v, u) \in E \) and \( u \in D \). Note that an edge \( (v, u) \) is, trivially, a path of length one between node \( v \) and node \( u \). This idea of a dominating set easily extends to the notion of \( k \)-domination. A set of nodes, \( D \subset V \), is called a \( k \)-dominating set if every node \( v \in V \) is either in \( D \), or is connected to a node in \( D \) by a path of length at most \( k \). Intuitively, it can be noticed that if the nodes corresponding to \( v \in D \) form the CHs in a system, these CHs can collect data from their local neighborhoods at the first tier and send the processed data to the collection point in the second tier. The problem of computing a dominating set and checking if a set is a dominating for a general graph is known to be NP-complete, which makes it computationally difficult.

To work around the inherent computational difficulty of the dominating set problem, this investigation focused on a class of regular mesh topologies that arise by embedding nodes in a 2D-BaseGrid and limiting the transmission range of each node using the embedded functions \( \Xi_q \), \( q = 3, 4, 6, 8 \) reported in [10]. In these topologies, each node has \( q \) neighbours. For each topology, an intuitive tiling pattern is presented. These patterns enable easy determination of \( k \)-dominating sets. It is shown that the parameter \( k \) affects the number of nodes in a cluster, the number of CHs, and hence the QoS and the energy efficiency of the data aggregation scheme [11]. Simulation results demonstrate the effectiveness of the proposed scheme.
1.1 Contributions

The specific contributions of this thesis are:

1. A system model that represents the energy efficiency vs. QoS tradeoff,

2. Tiling patterns for regular mesh topologies in which each node has $q$ neighbors,
   for $q = 3, 4, 6,$ and $8$ and increasing $k$-domination,

3. A comparative evaluation of the performance at the level of individual cluster heads and the system-level, and

4. A comparative analysis of $k$-domination Vs. QoS.

1.2 Overview of the Thesis

Chapter 2 presents the background necessary for the thesis. The system model that is used is presented and discussed in Chapter 3. Chapter 4 presents the tiling patterns that lead to $k$-domination sets in the mesh topologies generated by the embedding functions $\Xi_3, \Xi_4, \Xi_6,$ and $\Xi_8$. Simulation results are presented in Chapter 5. Finally, conclusions and future research steps are presented in Chapter 6.
CHAPTER II
BACKGROUND

Data aggregation is challenging when it is important to simultaneously achieve a high Quality of Service (QoS) because the metrics that quantify QoS are interdependent. For example, the energy efficiency of a protocol also critically impacts the latency and jitter in a system. Consequently, many data aggregation protocols reported in the literature are application specific. In contrast, this thesis presents an approach that is not application specific.

The approach to data aggregation reported in this thesis builds on prior results. The first important idea is that of $k$-domination in graphs. Nodes in the system are partitioned into subsets based on $k$ and one node in each partition is designated as a cluster head (CH). Nodes in a cluster send their messages to the CH, and the CH processes the received data and sends it towards a collection point (CP). This approach was evaluated using regular mesh topologies that are described next.

2.1 Regular Mesh Topologies

Mesh topologies considered in this investigation are obtained by embedding the nodes in a system, $N$, on a 2D-BaseGrid. Each location on 2D-BaseGrid is identified by a unique ordered pair $(i, j)$. The distance between two consecutive locations on the grid,
i.e., between \((i, j)\) and \((i, j + 1)\) or between \((i, j)\) and \((i + 1, j)\), is \(b\). An embedding function, \(\Xi_q\), assigns a location on 2D-BaseGrid and a transmission range to each node, \(n_i \in N\), such that each node embedded on the 2D-Basegrid has \(q\) neighbors [12].

\[
\Xi : n_i \in N \rightarrow (\mathbb{N} \times \mathbb{N} \times \mathbb{R}). \tag{2.1}
\]

Figure 2.1: Regular Mesh Topologies obtained using embedding functions \(\Xi_4\) and \(\Xi_3\) [12]

Figure 2.1 presents two examples. The figure on the left is the topology obtained using \(\Xi_4\) and the topology on the right is from \(\Xi_3\). Table 2.1 presents a list of 2D embedding functions. The regular mesh topologies obtained using the embedding functions are easier to analyze than general topologies, embedding functions for different mesh topologies are listed in Table 2.1. The QoS results based on such topologies represent upper bounds for what can be achieved in general topologies using similar techniques.
Table 2.1: Embedding functions that embed a collection of nodes $n_i \in N$ on a 2D-Basegrid with $R$ rows and $C$ columns [12].

<table>
<thead>
<tr>
<th>Row</th>
<th>Column</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Xi_3$</td>
<td>$2(i-1) \mod \left(\frac{C}{2}\right) + n^i_0$</td>
<td>$\left\lfloor \frac{i-1}{2} \right\rfloor - 1$</td>
</tr>
<tr>
<td>$n^i_0 = \begin{cases} 0 &amp; 0 \leq i \mod 4k \leq k, \text{ or} \ 3k + 1 \leq i \mod 4k \leq 4k - 1 \end{cases}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Xi_4$</td>
<td>$(i-1) \mod C$</td>
<td>$\left\lfloor \frac{i-1}{C} \right\rfloor$</td>
</tr>
<tr>
<td>$\Xi_6$</td>
<td>$4(i-1) \mod \frac{C}{4} + 1$</td>
<td>$\left\lfloor \frac{i}{4} \right\rfloor$</td>
</tr>
<tr>
<td>$4(i-1) \mod \frac{C}{4} + 3$</td>
<td>$\left\lfloor \frac{i}{4} \right\rfloor$</td>
<td>$b\sqrt{5}$</td>
</tr>
<tr>
<td>$\Xi_8$</td>
<td>$(i-1) \mod C$</td>
<td>$\left\lfloor \frac{i-1}{C} \right\rfloor$</td>
</tr>
</tbody>
</table>

2.2 Related Work

Since the device and link level characteristics of networked embedded systems are similar to sensor networks, data aggregation protocols designed for sensor networks are related to the approach proposed in this thesis. Such protocols reported in the literature focussed on improving QoS metrics, such as network lifetime, energy efficiency, latency, data accuracy, jitter, data freshness and message overhead per round. Many of these protocols are application specific and the requirements of the application impact the relevance of the QoS metrics. An excellent survey of data aggregation protocols for sensor networks is presented in [13].

Sensor network architecture critically affects the performance of the protocols. Protocols such as SPIN [7], Directed Diffusion [2], Push or Pull protocol [14] are examples that rely on flat networks [16]. There are disadvantages such as presence of duplicate messages, excessive communication at few nodes resulting in quick depletion of energy, and brittleness of the system.
Hierarchical architectures at the system level alleviate some of the above problems [13]. Nodes in such architectures are arranged in levels and aggregation techniques are applied at each level. This reduces overhead, energy consumption, improves scalability and energy efficiency. LEACH [4], PEGASIS [9], HEED [18] are protocols that rely on employ hierarchial architectures. Table 2.2 lists out the differences in hierarchial and flat networks.

**Table 2.2: Comparing Hierarchical and Flat networks.**

<table>
<thead>
<tr>
<th>Hierarchical Networks</th>
<th>Flat Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH processes data and sends aggregated data to collection points.</td>
<td>Individual nodes send data to collection points over multihop routes. Intermediate nodes may or may not process data.</td>
</tr>
<tr>
<td>System lifetime is not critically impacted by failure of a few nodes or CHs at lower levels of the hierarchy.</td>
<td>Failure of nodes that participate in many multihop routes can disrupt system.</td>
</tr>
<tr>
<td>Reduced network bandwidth and improved latency because CHs aggregate data.</td>
<td>Relatively large latency and jitter, no savings in bandwidth if data is not aggregated enroute.</td>
</tr>
<tr>
<td>The hierarchical structure allows one to quickly correlate data from areas or sections of the system.</td>
<td>System structure must be embodied in the multihop paths.</td>
</tr>
<tr>
<td>System lifetime can be extended by rotating nodes in a cluster to act as CHs.</td>
<td>Reconfiguring multihop paths requires additional protocols.</td>
</tr>
<tr>
<td>Cluster formation overheads during system initialization or restart. The hierarchical structure enables system-level topology debugging.</td>
<td>Multihop routes only span areas with successful transmission. Flat structure complicates system-level topology debugging.</td>
</tr>
<tr>
<td>Potential for improved energy efficiency.</td>
<td>Potential for improved robustness.</td>
</tr>
<tr>
<td>Scalable but reduces utilization with increased system size.</td>
<td>Scalability depends on efficiency of protocols.</td>
</tr>
</tbody>
</table>

2.2.1 Early Data Aggregation Protocols

Flooding is a simple solution in which every node sends all its data to all its neighbors. The nodes do not need to store any information about the topology. To avoid sending
messages repeatedly, duplicate messages are deleted from the system. This approach is not energy efficient and the latencies are large even when duplicate messages are eliminated in local neighborhoods.

A few interesting protocols for flat networks have been reported in the literature. Directed Diffusion [2] minimized data transmission and aimed to achieve robust efficient data distribution in network. This protocol is not useful for continuous data delivery applications. SPIN [7] is a negotiation based protocol that requires a factor of 3.5 less energy, distributes 60 percent more data compared to flooding. No global knowledge of the system topology required. Push or Pull [14] protocol uses anchor nodes that request other nodes to send data; the anchor nodes then determine the direction of data flow. The underlying mechanism is essentially flooding with duplicate elimination.

2.2.2 Data Aggregation Protocols exploiting Architectures

These protocols can be broadly classified as CH approaches and non-CH approaches. The protocols discussed in this section rely on architectural features such as infrastructure nodes, hierarchy, chains of nodes, trees of nodes, and structures of message flow to aggregate data.

The work in [5] used an aggregation tree rooted at the collection point. This tree is formed as the union of paths from different sources to the collection point and is called opportunistic tree aggregation. The greedy incremental tree [16] established a shortest path between the farthest nodes and collection point during initialization;
the remaining nodes connect themselves to the closest node on this tree. The protocol proposed in [16] compares no aggregation scenario with data aggregation conducted using methods such as, grid based, in network and hybrid. Results show that data aggregation improves performance of network significantly.

The first cluster based approach called LEACH was proposed in [4] and further improvements were reported in [4]. LEACH used an adaptive self-configuring cluster formation technique as the backbone for data aggregation. Data is aggregated in rounds. In every round, nodes organize themselves into local clusters, and one node in the cluster serves as a CH. PEGASIS [9], modified LEACH by exploiting a structure of nodes called chains. PEGASIS achieved 100 to 200 percent more efficiency than LEACH. Each node receives and transmits information to neighbors and takes turn to serve as a CH. Local coordination of nodes reduces bandwidth requirements and increases delay at the distant nodes in the chain. PEGASIS, however, assumed that all nodes can reach base station.

LRS [8] protocol was designed to improve the Energy Delay product metric - hence increased system lifetime and reduced latency. This protocol used two schemes - a binary scheme with CDMA based nodes and a 3-level hierarchical network. This approach outperformed LEACH by factor of 8 and PEGASIS by factor of 4. HEED [18] is another protocol that extended system lifetime. This was designed to handle large scale systems and provide robust aggregation. CHs were selected based on the residual energy remaining at the nodes, degree of the cluster head, power and density of the cluster. Only 1-hop communication was permitted to sensors. The pe-
periodic re-clustering ensured uniform depletion of energy among the nodes in a cluster. HEED terminated clustering process within constant number of iterations.

MLDA/CMLDA [3] protocols improved system lifetime. The underlying network was viewed as a directed graph $G$ with the root at the base station, and nodes in the graphs as the sensors. Directed trees were formed based on the location and energy of the sensor. Sensors aggregate data from neighbors and forward it to next node, and the messages are routed towards base station. CMLDA divided network with $N$ nodes to have $M$ cluster heads, and clusters were treated as super sensors and the these, in turn, communicated data to the sink.

Data Aggregation Rate [6] improved the system lifetime, by decreasing the number of packets sent from the CH as the aggregation rate increased. Dynamic Data Aggregation [1] used reconfigurable CHs by utilizing FPGA’s that were partially reconfigurable at run time. EAPS/hEAPS [17] was designed to have an optimal number of cluster heads in the network, and the cluster head selection depended on the probability assigned to each aggregator node. Optimum number of cluster heads were achieved. An $N \times N$ network was covered using aggregators with different probabilities. The protocol showed that total energy and maximum energy were reduced by factor 2 by using the hEAPS compared to EAPS.

Other protocols for data aggregation include [15, 19]. The use of $k$-domination to determine cluster heads is closely related to the EAPS [17]. The system model used in this investigation is discussed in the next chapter.
CHAPTER III
SYSTEM MODEL AND METRICS

This chapter presents a system model that was used in this investigation. Section 3.1
presents the organization of the system and Section 3.2 presents the data aggregation
scheme. The energy metrics that impact the viability of the cluster head based
approach are discussed in Section 3.3.

3.1 Nodes and Collection Points

Let $N$ represent the set of nodes in the system. Each node $n_i \in N$ corresponds to a
unique node $v_i \in V$, of the underlying graph, $G = (V, E)$. Recall that the embedding
functions, $\Xi_q$, assign a transmission range to the nodes $n_i \in N$. This transmission
range determines the neighbors $n_j \in N$ of the node $n_i$. For each such neighbor, $n_j$,
$(v_i, v_j)$ is an edge, $(v_i, v_j) \in E$, of the underlying graph $G$. We considered a single
collection point, $(CP)$. The data from all the nodes $n_i \in N$, must arrive at $CP$.
There is a distinguished node in $G$ that corresponds to $CP$.

There are one or more distinguished nodes, $\mathcal{C} \subset N$, that are called the cluster
heads (CHs). The set $\mathcal{C}$ is selected based on $k$-dominating sets of $G$ as discussed in
detail in the next chapter, i.e., Chapter 4.
3.2 Data Aggregation Scheme

We considered a three-tiered, two-level, system architecture in which the tiers are represented by

1. the nodes \( n_i \in N \),

2. the nodes \( c_i \in \mathcal{C} \), and

3. \( CP \), the collection point.

At the first level of the system, each CH\(^1\), \( c_i \in \mathcal{C} \), gathers data from the nodes \( n_i \in N \), via some path of length \( k \) in \( G \). \( c_i \) processes this data from its neighborhood and sends the aggregated data towards \( CP \) in the second level of the system. The nodes \( n_i \in N \) communicate with their cluster head, \( c_i \in \mathcal{C} \) via a multihop route that is selected using a standard shortest path algorithm.

It is assumed that all the nodes in \( N \), have an algorithm that is used to locally determine whether or not it is a CH. If a node determines it is a CH, it broadcasts a message to the neighboring nodes identifying itself as a CH. If a node \( n_i \) receives identifying messages from more than one CH, \( n_i \) uses link-level metrics such as Received Signal Strength Indicator or Link Quality Indicator \([13]\) to select one of the CHs. Once the CH is identified, the shortest path to the CH is also established.

Data aggregation proceeds in *rounds*. In each round, a CH gathers data from the nodes that are connected to it via some path of length \( k \). The CH processes this

\(^1\)Nodes in \( \mathcal{C} \) are distinguished nodes in \( N \). In this thesis, ”nodes in \( N \)” is used to refer to the nodes in \( \mathcal{C} \cup N \). When the distinction is important, \( c_i \) is used to refer to nodes in \( \mathcal{C} \). It is implied that for all \( n_i \in N \), \( n_i \not\in \mathcal{C} \).
data and prepares a packet that will be sent to CP. The CHs then send all the data, one from each CH, to CP at the second level of the system.

3.3 Energy Metrics

Let $e$ be the energy required for any node to send a byte of data to its immediate neighbor in the system. It is assumed that this cost is uniform across all the links in the system. For a node $n_i \in N$ that sends data to its CH, $c_i$, over a path of length $k$, $ke$ units of energy is expended for each data packet that is sent from $n_i$ to $c_i$. Let $E$ represent the energy required for any node, including the CHs, to send a byte directly with CP.

Consider a collection of CHs in which each cluster has $C_N$ nodes. The asymptotically best value of $H = |\mathcal{C}|$, i.e., the number of clusters, is

$$H = |N|/C_N$$  \hspace{1cm} (3.1)

Since the data aggregation proceeds in rounds, the energy expended by the nodes in each cluster, in each round (in the worst case) is

$$(E + k \times e \times p \times C_N)$$

Where $p$ is the number of bytes generated at each node. This is because, in each round, the CH must collect data from every node in its cluster to send this data to CP.
Suppose there were no cluster heads used. Then, the $C_N$ nodes in the above cluster must expend $p \times E \times C_N$ units of energy to send the same data to CP. Depending on the application, and the type of the data used, the CH aggregates the data it receives from members in its cluster to a reduced set of data items. Let $C$ represent the compression factor, i.e., the reduction factor in the volume of data at a CH, $C \in (0, 1]$. Thus, by considering such compression, the energy expended in each cluster, to send the aggregated data to the sink is

\[(1 - C) \times p \times E \times C_N + k \times p \times e \times C_N\]

Therefore the cluster based approach is viable only when

\[k \times e < E \times C\]

While this appears to be a reasonable relationship, the large difference between $E$ and $e$ make the values of $k$ impractical for pragmatic networks. The next chapter, presents a method for selecting $k$-dominating sets in regular mesh topologies when each node has $q = 3, 4, 6, \text{or } 8$ neighbors. By exploiting the structure of the topology, tighter bounds for the viability of the cluster based approach are presented.

\[^{2}\text{Since the cost of communication is at least an order of magnitude higher than the costs of computation in networked embedded systems, the cost of processing the data is ignored.}\]
CHAPTER IV
IDENTIFYING CLUSTER HEADS IN REGULAR MESH TOPOLOGIES

This chapter presents the approach to selecting CHs in regular mesh topologies. The nodes are assumed to be embedded in an infinite 2D-Basegrid as discussed in Chapter 2. The system of interest, i.e., the nodes that must be covered by the CHs, is arranged along $R$ rows and $C$ columns.

4.1 Definitions and Approach

Given a mesh topology $\Xi_q$, for a node $n \in N$, let $N^q_k[n]$ denote the closed $k$-neighborhood of $n$, i.e., the nodes $x \in N$ that are at distance $\leq k$ from $n$. Let $t^q_k = |N^q_k[n]|$. The results in this section demonstrate that the number of clusters is asymptotically $\frac{|N|}{t^q_k}$.

Initially, a maximal collection of vertices in $N$ is selected to be in $C$ such that no two vertices in $C$ are within $k$ hops of each other.

For a fixed $\Xi_q$, let $k$-tile$[n]$ be the subgraph induced by $N_k[n]$; this is the set of nodes in $N_k[n]$ and the edges between these nodes that are in the original topology. For each node $c_i \in C$, consider the $k$-tile$[c_i]$ - the cluster node $c_i$ is intuitively referred to as the center of the tile.
To construct a $k$-dominating set for $N$, the objective was to select a small number of vertices for $C$ such that the subgraphs induced by the $N_k(c_i)$ comprise vertex-disjoint $k$-tiles and all the nodes in $N$ are in some $k$-tile. Ensuring that the nodes in $C$ dominate vertices near the boundaries of the $R$ rows and $C$ columns of $N$ is difficult. Here, the presented approach departed from constructing a theoretically optimal dominating set.

Some of the $k$-tiles that lie at the boundary of $R$ rows and $C$ columns do not have the same number of nodes as other $k$-tiles - such tiles that lie at the boundaries are called $k$-semitiles in this thesis.

Let $\text{Cover}(C) = \bigcup_{c_i \in C} N_k[c_i], \forall c_i \in C$; this represents the nodes of $N$ that are in some $k$-tile. To ensure that $\text{Cover}(C)$ includes all the nodes in $N$, nodes $z_i \in N \setminus \text{Cover}(C)$ are associated with the CH of the closest semitile by breaking ties arbitrarily, i.e., a node $z_i$ is added to the closest semitile for which the CH at the center of the semitile is closest to $z_i$.

Mesh topologies that correspond to the embedding functions $\Xi_3, \Xi_4, \Xi_6, \text{ and } \Xi_8$ are different. For each of these topologies, the following sections present details of the dominating sets and tight bounds for the viability of the cluster based approach are also presented.

4.2 Minimal $k$-Dominating Sets

This section presents the method for identifying close approximations to the $k$-dominating sets that was used in the investigation.
As an example, first consider the mesh topology corresponding to $\Xi_4$. Each node has 4 neighbors; node $n_{i,j}$, i.e., the node that is at the intersection of the $i^{th}$ row and the $j^{th}$ column, is adjacent with nodes $n_{i+1,j}$, $n_{i-1,j}$, $n_{i,j+1}$, and $n_{i,j-1}$. Figure 4.1 shows an example in $\Xi_4$ when $R = C = 10$. Each diamond in the center of the figure is a 1-tile. The center node of each tile is intuitively clear from the picture. Note that some of the tiles at the boundaries are shown with dotted lines - these are the semitiles. The dark nodes are not in any tile or semitile; these nodes are associated with the CH of a semitile that is closest to the node.

Figure 4.1: The diamonds represent the tiles and, hence, $\text{Cover}(\mathcal{C})$ of $N$ in $\Xi_4$. The $k$-semitiles at the boundaries are shown with dotted outlines.
Consider the set of nodes $D = n_{i,j} | j = i(2k + 1)(\text{mod } t_k)$. $D$ is a $k$-dominating set of $N$ in $\Xi_4$ with the property that the set $\{N_k[v] | v \in D\}$ is a partition of the vertices of $N$. Consider $\text{Cover}(D)$ and the vertices in $N$ that are not $k$-dominated by some vertex in $D$. Because the tiles that are centered at the vertices in $D$ partition $N$, the vertices in $N \setminus \text{Cover}(D)$ can be precisely identified. For $\ell \leq k$, an $(\ell, k)$-semitile is the graph composed of the bottom $\ell$ rows of a $k$-tile. Each row of a tile, and hence each row of a semitile, has an odd number of vertices. Hence, the center of an $(\ell, k)$-semitile $T$ is the center vertex in the top row of $T$. Thus each vertex in $N \setminus \text{Cover}(D)$ will lie in $(\ell, k)$-semitiles centered at a vertex $v$ on the boundary of $N$. Each of these semitiles can be uniquely associated with a tile centered at a vertex in $\text{Cover}(D)$ whose center lies outside of $N$. The center of each of these semitiles is added to $D$ to identify the the CHs for $N$. Figure 4.2 presents the tiles, semitiles and covers in the mesh topologies $\Xi_8$ and $\Xi_6$ for $k = 1$.

![Figure 4.2: Cluster Head selection in $\Xi_8$ (left) and $\Xi_6$ (right) based on 1-domination.](image-url)
Table 4.1 is a summary for the nodes that must be selected as the initial set of nodes for $D$ in various mesh topologies for different $k$. Note that the tiling patterns for $k$-domination in $\Xi_3$ are not similar when $k$ changes as shown in Figure 4.3 and Figure 4.4.

Table 4.1: In mesh topologies $\Xi_3$, $\Xi_4$, $\Xi_6$, and $\Xi_8$, select nodes $n_{X,Y}$ as cluster heads as the initial set of nodes for the set $D$.

<table>
<thead>
<tr>
<th>Topology</th>
<th>X</th>
<th>Y</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Xi_3$</td>
<td>$i(2k)$</td>
<td>$j(2k + 2)$</td>
<td>$\forall i(2k) \leq R$, $j(2k + 2) \leq C$</td>
</tr>
<tr>
<td></td>
<td>$i(2k) + 1$</td>
<td>$j(2k + 2)$</td>
<td>$i \in {1, 3, 5, 7, \ldots}$, $\forall i(2k) + 1 \leq R$, $j(2k + 2) \leq C$</td>
</tr>
<tr>
<td>$\Xi_4$</td>
<td>$i(2k + 1) \mod t_k^4$</td>
<td>$i$</td>
<td>$\forall i(2k + 1) \mod t_k^4 \leq R$, $i \leq C$</td>
</tr>
<tr>
<td>$\Xi_6$</td>
<td>$i(2k + 1) \mod t_k^6$</td>
<td>$i$</td>
<td>$\forall i(2k + 1) \mod t_k^6 \leq R$, $i \leq C$</td>
</tr>
<tr>
<td>$\Xi_8$</td>
<td>$i(2k + 1)$</td>
<td>$j(2k + 1)$</td>
<td>$\forall i(2k + 1) \leq R$, $j(2k + 1) \leq C$</td>
</tr>
</tbody>
</table>

Figure 4.3: Tiling pattern for 1-domination.
Figure 4.4: Shaded vertices as cluster heads for vertices not 2-dominated in $\mathbb{Z}$ using shortest path algorithm.

One of the advantages of the dominating sets constructed here is that the associated clusters are disjoint. This eliminates the need for a node lying in multiple clusters to undertake any decision about which CH to associate with. Additionally, this method implies that the process of selecting the CHs can be greatly simplified, as compared to other domination-based CH selection algorithms, when finding a $k$-dominating set in these regular meshes. This is advantageous as at various times, we may wish to quickly aggregate the data in a given contour lying within a larger mesh topology.
4.3 Viability of Clusters

Consider a $k$-dominating set for the system with $N$ nodes. Each member of this set is a CH. Each cluster has a total number of $C_N = q.(k^2 + k)/2$ nodes with $qk$ nodes at a distance $k$ from a cluster head. Let $H$ denote the number of clusters; $H = N/C_N$ is the asymptotically best value for $H$.

The energy required for $R$ data aggregation rounds, i.e., to collect data from each node $R$ times, using CHs is $R(E + \sum_{j=1}^{k} q.j(e_j))$ units.

Suppose there were no cluster heads used. Then, the nodes in the above cluster require a total of $R(E.(q.(k^2 + k)/2 + 1))$ units of energy for $R$ rounds. Depending on the application, and the type of the data used, the CH aggregates the data it receives from members in its cluster to a reduced set of data items. Let $C$ represent the compression ratio, i.e., the reduction factor in the volume of data at a CH. Thus, by considering such compression, the energy required for a CH to send the aggregated data to the sink, in the worst case, is $R((1 - C)E.C_N + \sum_{i=1}^{k} q.i.e_i))$.

4.3.1 Viability of Clusters in $\Xi_4$

The energy saved by using the $k$-domination based CH approach in $\Xi_4$ mesh topologies is

$$(E.(4.(k^2 + k)/2 + 1)) - (1 - C)E.C_N + \sum_{i=1}^{k} 4.i.e_i) > 0.$$

$$E.(1 + 4ck(k + 1)/2) - \sum_{i=1}^{k} 4.i.e_i) > 0$$

$$(E(1 + c.4.k.(k + 1)/2) - 4.e.k.(k + 1).(2k + 1)/6 > 0$$
Thus, we need
\[ \frac{2k + 1}{3\left(\frac{1}{2k(k+1)} + C\right)} < \frac{E}{e}. \] (4.1)

4.3.2 Viability of Clusters in Ξ₃

The energy saved by using the \( k \)-domination based CH approach in Ξ₃ mesh topologies is
\[ (E.(3.(k^2 + k)/2 + 1)) - ((1 - C)E.C_N + \sum_{i=1}^{k} 3.i(i.e)) > 0. \]
\[ (E.(1 + 3ck(k + 1)/2) - \sum_{i=1}^{k} 3.i(i.e) > 0 \]
\[ (E(1 + c.3.k.(k + 1)/2) - 3.e.k.(k + 1).(2k + 1)/6 > 0 \]

Thus, we need
\[ \frac{2k + 1}{(\frac{2}{k(k+1)} + 3C)} < \frac{E}{e}. \] (4.2)

4.3.3 Viability of Clusters in Ξ₆

The energy saved by using the \( k \)-domination based CH approach in Ξ₆ mesh topologies is
\[ (E.(6.(k^2 + k)/2 + 1)) - ((1 - C)E.C_N + \sum_{i=1}^{k} 6.i(i.e)) > 0. \]
\[ (E(1 + c.6.k.(k + 1)/2) - 6.e.k.(k + 1).(2k + 1)/6 > 0 \]

Thus, we need
\[ \frac{2k + 1}{\frac{1}{k(k+1)} + 3C} < \frac{E}{e}. \] (4.3)
4.3.4 Viability of Clusters in $\Xi_8$

The energy saved by using the $k$-domination based CH approach in $\Xi_8$ mesh topologies is

$$(E.(8.(k^2 + k)/2 + 1)) - ((1 - C)E.C_N + \sum_{i=1}^{k} 8.i(i.e)) > 0.$$  

$$(E(1 + c.8.k.(k + 1)/2) - 8.e.k.(k + 1).(2k + 1)/6 > 0$$

Thus, we need

$$\frac{2k + 1}{3(\frac{1}{4k.(k+1)} + C))} < \frac{E}{e}. \quad (4.4)$$

Note that in Equations (4.1), (4.2), (4.3), and (4.4) there is a small fractional term involving the parameter $k$ in the denominator. For example in Equation (4.4), this term is $1/(4k.(k + 1))$. By ignoring these small terms, the $k$-domination based CH approach is viable when

$$\frac{(2k + 1)}{3C} < \frac{E}{e}. \quad (4.5)$$

This means that for a given set of values of $E$ and $e$ that are obtained either from manufacturer data sheets or measurements, and the nature of the application that dictates the compression ratio $C$, this approach is viable for a value of $k$ when

$$k < \frac{3.C.E}{2.e} - \frac{1}{2}. \quad (4.6)$$
CHAPTER V

RESULTS

To evaluate the QoS achieved using $k$-domination based CH approach, a discrete event simulator was designed and implemented using the OMNet++ framework.

5.1 QoS Metrics

In each simulation run, the number of messages handled by each node in the cluster and the number of messages the node generated were counted. The times at which each message was sent from the node in the cluster and arrived at the CH were recorded. The time at which the aggregated message was sent by CH and arrived at sink was recorded. The difference between these two times was recorded as the end-to-end latency for the message for the CH under consideration. The average latency, jitter, message loss rate, over all the messages at the CH was computed for each cluster. The throughput at system level, which was defined as number of messages arriving at the collection point per unit time-was computed. Throughput was also computed at every cluster by counting the number of messages arriving at cluster head per unit time.
5.2 Simulation Approach

The multihop, multipath propagation of messages from the nodes to the cluster head was modeled as a multistage queuing network. The service time of each queue was exponentially distributed with a mean time of 250 ms. This time represents the time required to propagate a message from the preceding node, the time required to receive a message in the node, and time required to forward the message towards the sink. Using the channel modeling capability in OMNet++, the data rate (bandwidth) was set to 38.4kbps. Propagation delay and bit error rates were assumed to be zero, which are the default values in OMNet++. All the links between every pair of nodes in the wireless mesh topology used the same channel model. The inter-arrival time of the messages generated at nodes was exponentially distributed with a mean time of 150 ms.

The above QoS metrics were used as the basis for comparing the performance levels achieved by clusters with varying $k$-domination in $\Xi_3$, $\Xi_4$, $\Xi_6$ and $\Xi_8$. These results are used to evaluate which viable $k$-dominations for pre designed QoS constraints. The results reported are based on the scenario in which each node sends 100 messages toward its cluster head and had a queue length of 10 messages. 10x10 and 20x20 size grids are simulated to study the effects of large networks. The nodes in each cluster route their messages using the shortest paths towards the cluster head. The messages received by the cluster head are aggregated with the predetermined compression ratio and sent to the sink. Multihop communication networks are formed
when the value if $k$ increases, as the number of nodes routing messages towards the cluster head increases.

5.3 Effect of $k$ on QoS

QoS metrics are compared against $k$ values, which are used to evaluate $\Xi_3$, $\Xi_4$, $\Xi_6$, and $\Xi_8$. Latency in $\Xi_4$ for $k = 2$, shown in Figure 5.1 is less compared to the latency observed when $k = 5$ in Figure 5.2. The above result shows the effect of multihop network, where messages take more time to reach the cluster head in $k = 5$, compared to $k = 2$.

![Figure 5.1: Latency in $\Xi_4$ with $k=2$ on a 20x20 grid.](image)

Jitter observed in $\Xi_6$ on a 20x20 grid for $k = 1$, Figure 5.3, is much less compared to the jitter seen for $k = 5$ from Figure 5.4. The observed performance results because the congestion builds up as the number of paths towards the cluster
head increase, and also because of the increased shortest path length from $k = 1$ to $k = 5$.

Effect of $k$ on throughput is not predominant, the throughput observed when $k = 1$ in Figure 5.5 is the maximum achievable because of the single hop network around the cluster head, and throughput does not reduce significantly as the network around the cluster head grows as $k = 5$ seen in Figure 5.6.

Figure 5.7 and Figure 5.8 showed the increased percentage message loss rate, as congestion builds with large values of $k$.

The above results compared QoS metrics with different $k$; understanding the performance of different mesh topologies is essential to decide on which mesh topology to choose for required QoS. In Figure 5.9 $\Xi_3$ and $\Xi_6$ achieved better latencies compared to $\Xi_4$ and $\Xi_8$, for $k = 1$, because less number of clusterheads and low congestion, this
performance trend repeats even at high value of $k = 3$ as seen in Figure 5.10.

The above studies helped us to understand the performance of the clustering technique at the cluster head level, but a clear understanding of the performance at the sink level is essential to decide which mesh topology and what value of $k$ needs to be chosen.

5.3.1 Effect of $k$

To compare the performance of $\Xi_3$, $\Xi_4$, $\Xi_6$, and $\Xi_8$, domination numbers from $k = 1$ to $k = 5$ are simulated on 10x10 and 20x20 grids.

Number of nodes in a cluster and the number of hops required by nodes in the cluster effect the performance at both cluster head and sink level, Figure 5.11 shows that message loss rate increases as the value of $k$ increases. The nodes near
the cluster head handle more number of messages compared to the ones farther away, which results in queue build up and overflow resulting in message loss.

Stable performance in $\Xi_3$ is provided by the less dense node structure and which result in less number of nodes in each cluster. Figure 5.12 points out that value of $k$ has very less effect on the message loss rate in $\Xi_3$.

Figure 5.13 and Figure 5.14 show that message loss rate is minimum in $\Xi_3$ on 10x10 compared to maximum loss observed in $\Xi_4$, this performance trends flips on a 20x20 grid where $\Xi_4$ achieve minimum losses compared to maximum observed in $\Xi_3$.

5.3.2 Performance at Sink Level

Jitter observed in $\Xi_3$ stays stable with out significant increase or decrease as the $k$ increases which is observed Figure 5.18.
With large $k$ values number of nodes sending messages to sink decreases, and number of messages increases as the more messages are received by the cluster head. Eventually the average latency observed at the sink will drop, as the messages are frequently sent towards the sink by cluster heads, the above performance can be observed for $\Xi_3$ and $\Xi_6$ in Figure 5.19.

Throughput achieved at the sink drops as the number of cluster heads which report to the sink decreases. Latency at the cluster head increases as the value of $k$ increases, so as the time taken by the cluster head to report to the sink, which results in lesser throughputs at the sink, which is shown in Figure 5.20.
5.3.3 Summary and Discussion

QoS performance in $\Xi_3$, $\Xi_4$, $\Xi_6$, and $\Xi_8$ decrease with increase in $k$. The observed performance resulted because of increase in number of nodes in each cluster and hops required to reach a cluster head.

Unpredictable behavior in $\Xi_6$ is accounted for less number of nodes for a given grid size. Overall performance in $\Xi_3$ is better compared to $\Xi_4$, $\Xi_6$, and $\Xi_8$. We can get less jitter, latency but at the expense of low throughput. The throughput observed is high in $\Xi_4$ and $\Xi_8$, because of the dense node structure, which account for more number of number received by the cluster head. This acts negatively on jitter, latency and loss rate as the nodes near the cluster head handle more number of messages resulting in queue buildup.
Figure 5.7: Percentage message loss in $\Xi_3$ for $k=2$ on a 20x20 grid.

For future work, the routing technique may be used for transmission of information from cluster heads to sink which may help to minimize the total energy of system and data loss. These topologies can be explored in different framework to nullify of OMNeT+ framework limitations.
Figure 5.8: Percentage message loss in $\Xi_3$ for $k=5$ on a 20x20 grid.

Figure 5.9: Latency comparison between $\Xi_3$, $\Xi_4$, $\Xi_6$, and $\Xi_8$ for $k=1$. 

34
Figure 5.10: Latency comparison between $\Xi_3$, $\Xi_4$, $\Xi_6$, and $\Xi_8$ for $k=3$.

Figure 5.11: Percentage message loss in $\Xi_4$.  

35
Figure 5.12: Percentage message loss in $\Xi_3$.

Figure 5.13: Percentage message loss with $k = 2$ on a 10x10 grid.
Figure 5.14: Percentage message loss with $k = 2$ on a 20x20 grid.

Figure 5.15: Average Latency and Jitter for $\Xi_4$, minimum latencies and jitter are observed when $k = 1$, because of the single hop network around the cluster head compared to the dense networks at higher $k$’s. The outlier performances observed is because of the semi-tiles in the grid.
Figure 5.16: Average throughput for $\Xi_8$, the throughput decreases as the number of nodes in the cluster increase.

Figure 5.17: Throughput comparison with $k=2$ and $k=5$. Higher throughputs are achieved by topologies which contribute more messages to the cluster heads in a short time like $\Xi_4$ and $\Xi_8$. 
Figure 5.18: Jitter at Sink level for $\Xi_3$

Figure 5.19: Latency at Sink level for $\Xi_3$ and $\Xi_6$
Figure 5.20: Throughput at Sink level for $\Xi_4$
CHAPTER VI

CONCLUSION

This thesis proposed a systematic approach to select the cluster heads for data aggregation in regular mesh topologies based on the idea of $k$-dominating sets in graphs. Mesh topologies in which each node had 3, 4, 6, and 8 neighbors were considered. For each topology, an intuitive tiling pattern was presented. A subset of nodes, called cluster heads, which approximate a $k$-dominating set of a graph, was identified for each mesh topology.

It was shown that three parameters impact the viability of the $k$-domination based approach to cluster head selection. If $E$ is the energy required for each node to communicate directly with a collection point, $e$ is the energy required for a node to communicate with its immediate neighbor, and $C$ represents the ratio of data compression achieved at the cluster heads, then the $k$-domination approach presented in this thesis was shown to be viable when $k < \frac{3CE}{2e} - \frac{1}{2}$. The results in Chapter 4 showed that while there are small differences in this constraint for different mesh topologies, the above constraint dominates when the compression ratios and the difference in energy, i.e., $|E - e|$, is high.
The parameter $k$ controlled a critical tradeoff between the performance at the
cluster level and that at the system level. The performance of the data aggregation
time was quantified in terms of jitter, latency, average throughput, and percentage
of message loss as the number of nodes per cluster increased. Simulation results
showed that mesh topologies in which each node had 4 neighbors or 8 neighbors
provided better throughput but with increased jitter and latency. In mesh topologies
in which each node had 3 neighbors, average jitter and latency remained constant
with increasing $k$, however, the throughput decreased. In the future, this work can
be extended to general mesh topologies and peer-to-peer systems.
BIBLIOGRAPHY


44