Domination and Coverage Guarantees Through Distributed Computation

D. Subhadrabandhu, F. Anjum, S. Kannan and S. Sarkar

Abstract

Node domination and coverage problems form the basis of several important resource allocation problems in communication networks and information systems. The problems are invariably NP-hard. Worst case approximation guarantees can however be obtained through existing polynomial complexity approximation algorithms. But, the algorithms known to attain these worst case guarantees are all centralized, and hence not suitable for use in large scale networks. We consider a simple distributed algorithm, MUNEN ("Maximum Unsatisfied Neighbors in Extended Neighborhood") which selects nodes based on a local selection algorithm at each node, and rigorously prove that it attains the best attainable worst case approximation guarantee in several versions of the node domination problem. Furthermore, we prove that in all these problems, MUNEN selects exactly the same nodes as a centralized approximation algorithm, Greedy, that attains the above worst case approximation guarantee by using network-wide information while selecting nodes. We next generalize MUNEN for node coverage problems. We prove that this generalization again attains the best attainable worst case approximation guarantee, and also selects the same set of nodes as the best known existing centralized algorithm, while substantially reducing the need for network-wide communications.

I. Introduction

A dominating set in a network is a set, D, of nodes such that every node in the network is at most one hop away from some node in D (Fig. 1). A minimum dominating set is a dominating set of minimum cardinality. Since every node in the network is at most one hop away from some node in a dominating set, nodes in a dominating set can effectively compute routes (e.g., using OLSR protocol), schedule transmissions and serve as mirror sites. Due to the broadcast property of the wireless medium, nodes in a dominating set of a wireless network can collectively receive all transmissions, and can therefore monitor the network for securing it, discover network utilities, and also aggregate sensed data in a sensor network. Selecting a minimum dominating set significantly reduces the resource consumed in discharging the above functionalities. Thus, efficient computation of minimum dominating sets forms the basis of several important resource allocation problems in communication networks.

Unfortunately, computing a minimum dominating set has been proven to be NP-hard [2]. But, well-known approximation algorithms exist that approximate the optimal solution within a multiplicative factor of $\Omega(\log(N))^*$, where N is the number of nodes in the network. It has also been proven that unless $P = NP$, all polynomial complexity algorithms will have approximation ratio of at least $\Omega(\log(N))$ in the worst case [2]. Several approximation algorithms have been proposed that attain a worst case approximation ratio of $O(\log(N))$.

D. Subhadrabandhu and S. Sarkar are at the department of Electrical and Systems Engineering, S. Kannan is at the department of Computer Science at the University of Pennsylvania. F. Anjum is at the Telcordia Technologies. Their emails are dhanant@seas.upenn.edu, fanjum@telcordia.com, kannan@cis.upenn.edu and swati@seas.upenn.edu. The contribution of S. Sarkar was in part supported by the National Science Foundation under grants ANI-0106984, NCR-0238340 and CNS-0435306.

*A function $f(N)$ is said to be $\Omega(N)$ if there exists a positive constant c such that for all large N $f(N) \geq cN$.

But, till date, all approximation algorithms that are known to provide worst case approximation guarantees ($O(\log(N))$ or any other meaningful bound) are centralized, and hence not suitable for use in large scale networks.

We consider a simple distributed algorithm, MUNEN ("Maximum Unsatisfied Neighbors in Extended Neighborhood") which computes a dominating set based on a local selection algorithm at each node (Section II). Specifically, a node selects itself if and only if it is the node with the maximum number of uncovered neighbors in its $2-$hop neighborhood. Given its simplicity and distributed nature of operations, MUNEN is likely to be widely deployed in resource constrained wireless networks provided some guarantees can be obtained on its performance. Our contribution has been to prove that MUNEN, not only attains the $O(\log(N))$ approximation ratio, but also selects exactly the same nodes as a centralized approximation algorithm, Greedy, that attains the above approximation ratio by using network-wide information while selecting nodes. Our result is somewhat counter-intuitive as it establishes that the use of local information does not penalize the performance in this case. We also prove that a further simplification of MUNEN, in which a node selects itself if and only if it has the maximum number of uncovered neighbors in its $1-$hop neighborhood, attains an arbitrarily poor approximation ratio. Thus, there seems to exist a "threshold" phenomenon regarding the extent of local information needed for attaining the best attainable approximation ratio.

We next consider several important generalizations of the minimum dominating set problem (Section IV). Coverage with multiple nodes may provide redundancy in coverage which is useful when the covering nodes discharge their responsibilities only periodically due to frequent failures or power-saving operations, e.g., in sensor networks. Different nodes may also need different amount of coverage redundancy. A *multi-cover* is a set M of nodes such that every node in the network is covered by its required number of nodes in M (Fig. 1). Another parameter is the *required coverage distance* which also depends on the network and the role of the covering nodes. For example, if the role of the covering nodes is to provide specific network services like serving as web-caches, databases or servers, then it may be sufficient to select a k-dominating set D_k which satisfies the property that every node in the network is within k hops of some node in D_k (Fig. 1). In all these cases, computing a minimum cardinality multi-cover or a k-dominating set will minimize the resource consumption subject to providing the desired coverage. Both computation problems are however NP-hard, and all approximation algorithms that are known to approximate the optimal solution within guaranteeable approximation bounds are centralized. We prove that MUNEN can be generalized to select in a distributed manner exactly the same set of nodes as the best known centralized approximation algorithm in each case. Thus, the best known approximation guarantees can be provided through distributed computation.

We next consider the "maximum coverage" problem which is the dual of the minimum dominating set problem in that it maximizes the number of nodes covered by a selected set subject to selecting no more than k nodes in the set (Fig. 1) (Section V). If for example the selected set is used in securing the network, this selection criterion minimizes the security risk subject to using no more than a certain fraction of network resources in securing the network. Again, it is well-known that the maximum-coverage problem is NP-hard. But, a generalization of Greedy (G-Greedy) uses an iterative computation to approximate the coverage attained by the optimal solution within a multiplicative factor of $1 - 1/e$. It is also known that unless $P = NP$, no polynomial complexity algorithm can be guaranteed to have approximation ratio greater than $1 - 1/e$ in every network [2]. G-Greedy however consumes k iterations and requires network-wide communication after every iteration. We generalize MUNEN to select the same set of nodes as G-Greedy, while also consuming k iterations, but requiring only 3 network-wide communications irrespective of k and the

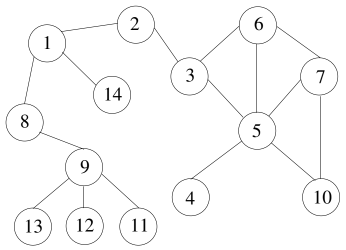

Fig. 1. We consider an example of a network G with 14 nodes. All nodes have equal weights. Both $\{1, 2, 4, 7, 9\}$ and $\{1, 5, 9\}$ are dominating sets in G. The minimum dominating set is $\{1, 5, 9\}$. All nodes are within transmission range of some node in this set. Thus, if G is a wireless network, nodes in this set collectively receive all network transmissions. The multi-cover $\{1, 2, 3, 4, 5, 9\}$ covers node 1 with 2 nodes, node 5 with 3 nodes (i.e., with 3, 4, 5) and the rest of the nodes with 1 node each. Here, $\{1, 3, 8\}$ is a $2-$dominating set. The set $\{5, 9\}$ covers 11 nodes ($\{3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}$) and maximizes the coverage among all node sets of cardinality 1 or 2.

topology. Thus, MUNEN attains the best known approximation guarantee of $1 - 1/e$ while exchanging significantly fewer messages as compared to existing approximation algorithms with worst-case performance guarantees.

We summarize our main contributions in Section VI.

II. Approximating Minimum Weighted Dominating Set Selection Through Distributed Computation

We investigate performance guarantees obtained by distributed algorithms for approximating the minimum dominating set computation. We first define some terminology, and briefly describe an existing centralized algorithm, Greedy, that is guaranteed to attain the best possible approximation ratio of $O(\log(N))$. We subsequently describe a simple distributed approximation algorithm MUNEN that selects nodes that are locally optimum in their 2-hop neighborhood. We prove that MUNEN selects the same set of nodes as Greedy, and therefore attains the best possible approximation ratio of $O(\log(N))$. We next consider a simplification of MUNEN, MUN ("Maximum Unsatisfied Neighbors") that selects nodes that are locally optimum in their 1-hop neighborhood. We prove that the performance of MUN can become arbitrarily bad, and has the same order as that of the trivial dominating set selection algorithm that selects all nodes.

We consider an undirected network $G = (V, E)$, where $V = \{1, \ldots, N\}$ and E respectively denote the sets of nodes and links. Every node u has a weight w_u which is a positive real number. The weights of different nodes may be different. The *weight of a set S of nodes* $W(S)$ is the sum of the weights of nodes in S.

Definition 1: A node u is a *neighbor* of or *covers* a node v if either (i) $(u, v) \in E$ (ii) or $u = v$. The *neighborhood* N_u of a node u is the set of its neighbors. The *degree* of a node u is $|N_u| - 1$. Let d be the maximum degree of a node in V.

By this definition, a node is always its own neighbor and covers itself.

Definition 2: A node u is an *extended neighbor* of a node v if (i) $u = v$ (ii) or $(u, v) \in E$ (iii) or there exists a node w such that $(u, w), (w, v) \in E$. The *extended neighborhood* EN_u of a node u is the set of its extended neighbors.

Thus, 2 nodes are neighbors (extended neighbors) if they are within 1 (2) hop of each other. For example, in Fig. 1, $N_1 = \{1, 2, 8, 14\}$ and $EN_1 = \{1, 2, 3, 8, 9, 14\}$.

Definition 3: A *dominating set* D in G is a set of nodes such that every node in V is covered by a node in D. A *minimum weighted dominating set* $D_{\min}$ is a dominating set that has the minimum weight among all dominating sets.

For example, in Fig. 1, let nodes $1, 5, 9$ have weights $4, 6, 5$ respectively, and let the rest of the nodes have weight 1. The minimum weighted dominating set $D_{\min}$ is $\{3, 4, 7, 8, 11, 12, 13, 14\}$.

Definition 4: The *approximation ratio* $\gamma_\pi(N)$ of a selection strategy π is the worst (maximum) possible value of the ratio between the weights of the dominating set selected by π and the minimum weighted dominating set in networks with N nodes.

We next state some well-known results. Let harmonic function $H(x) = \sum_{i=1}^{x}(1/i)$. Now, computing a minimum weighted dominating set, or even a dominating set whose weight is guaranteed to be less than $H(N)W(D_{\min})$ in any network with N nodes, is an NP-hard problem [2]. Note that $H(N) = O(\log(N))$. Thus, in this sense, $O(\log(N))$ is the best possible approximation ratio. Finally, computing a minimum dominating set or a dominating set with minimum cardinality is clearly a special case of computing a minimum weighted dominating set, and can be obtained by selecting equal weights for all nodes. The above hardness results hold in this special case as well.

We now describe an existing centralized approximation algorithm, Greedy, which attains an approximation ratio of $H(d+1)$ [2]. (Note that $d+1 \leq N$, and in the worst case $d+1 = N$. Thus, the approximation ratio of Greedy is $H(N)$ or $O(\log(N))$).

Definition 5: Let $A \subseteq V$. A node u is *satisfied under* A if it is covered by a node in A, and *unsatisfied under* A otherwise. Let $US_A(u)$ be the number of neighbors of u that are unsatisfied under A.

Definition 6: Let $A \subseteq V$. Let a node u be *globally optimum under* A if it has the minimum $w_u/US_A(u)^\dagger$ in the network. If multiple nodes attain this minimum, the one amongst them that has the minimum identity‡ is considered to be globally optimum.

For example, in Fig. 1, consider $A = \{1, 5\}$ and nodes $8, 9$. Here, $N_8 = \{1, 8, 9\}$. Clearly, node 1 in A covers both 1 and 8, but node 9 is still unsatisfied under A. Here, $US_A(8) = 1$. Also, $US_A(9) = 4$, and node 9 is globally optimum under A.

Greedy iteratively augments a set which is initially empty, so as to eventually obtain a dominating set. We denote the set of nodes selected by Greedy just before iteration t as $GD(t)$. While at least one node is unsatisfied under $GD(t)$, Greedy obtains $GD(t+1)$ by adding the node that is globally optimum under $GD(t)$ to $GD(t)$. It terminates whenever all nodes are satisfied under the selected set. For example, in Fig. 1, Greedy selects nodes $5, 9, 1$ in iterations $1, 2, 3$ respectively and terminates after completing 3 iterations. Clearly, Greedy terminates in at most N iterations as it selects a new node in each iteration.

Greedy is centralized as in each iteration it requires network-wide information to determine which node is globally optimum. Thus, a natural first step towards developing a distributed approximation algorithm is to select a *locally optimum* node instead of the globally optimum node in each iteration. We now consider a distributed approximation algorithm, MUNEN that selects nodes that are locally optimum in their extended neighborhoods. We first define the concept of local optima or "extended neighborhood optima."

Definition 7: Let $A \subseteq V$. A node u is *extended neighborhood optimum under A* if for every other node v in its extended neighborhood EN_u (i) either $w_u/US_A(u) < w_v/US_A(v)$ (ii) or $w_u/US_A(u) = w_v/US_A(v)$ and $u < v$.

For example, in Fig. 1, consider $A = \phi$. Both 5 and 9 are extended neighborhood optimum, while only 5 is globally optimum, under A.

Like Greedy, MUNEN starts with an empty set which progressively increases in size as nodes decide to join the set. We assume that time is slotted. We denote the cumulative set of nodes selected by MUNEN just before time t as $MD(t)$. Under MUNEN, a node selects itself in slot t if and only if it is extended neighborhood optimum under $MD(t)$. Refer to Figure 2 for a formal description of MUNEN. For example, in Fig. 1, $MD(1) = \{5, 9\}$, $MD(2) = \{1, 5, 9\}$ and $MD(i) = MD(2)$ for all $i > 2$.

†If $u \in A$ $US_A(u) = 0$ as all u's neighbors are satisfied, and hence $w_u/US_A(u)$ is infinite. Unless all nodes are satisfied under A, at least one node u has finite $w_u/US_A(u)$.

‡Note that nodes are represented by integers.

Fig. 2. Pseudo-Code for MUNEN

Let $MD(t)$ be the set of nodes selected under MUNEN just before time t.
$MD(1) = \phi$ /* No node selects itself before slot 1 */
Information at each node u at time t

begin

Extended neighborhood of u, EN_u.
The ratio $w_v/US_{MD(t)}(v)$ for each v in EN_u.

end

Procedure Decision at node u in slot t

begin

Node u selects itself if and only if (a) it is extended neighborhood optimum under $MD(t)$ and (b) $US_{MD(t)}(u) > 0$.

end

We now describe the analytical guarantees of MUNEN.

Theorem 1: The set $MD(N+1)$ selected by MUNEN at the end of N slots is the same as that selected by Greedy when Greedy terminates. $MD(t)$ converges in at most N slots.

The first part of the result is counter-intuitive due to the following major differences between MUNEN and Greedy. MUNEN selects all extended neighborhood optimum nodes in each slot, while Greedy selects only the globally optimum node in each slot. Thus, Greedy and MUNEN may select different number of nodes in the same iteration, and may also select nodes in different order which may in turn affect the optimality evaluations. The proof for Theorem 1 is therefore involved, and we consider it in a separate section, Section III.

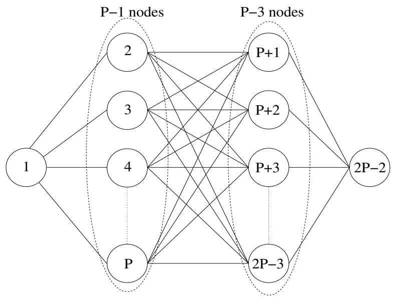

Fig. 3. Let $S_1 = \{2, \dots, P\}$ and $S_2 = \{P+1, \dots, 2P-3\}$ denote the nodes in the 2nd and 3rd columns respectively. Here, $N = 2P-2$ and $E = \{(1,2), \dots, (1,i), (i,j), (j, 2P-2),\ \forall\ i \in S_1,\ j \in S_2\}$. All nodes in the network have unit weight. Thus, $D_{\min} = \{1, 2P-2\}$, and $W(D_{\min}) = 2$. Let $A(t)$ denote the set of nodes selected by MUN just before slot t. $A(1) = \phi$. Thus, $US_{A(1)}(1) = P$, $US_{A(1)}(u) = P-1\ \forall\ u \in S_1$, $US_{A(1)}(v) = P+1\ \forall\ v \in S_2$ and $US_{A(1)}(2P-2) = P-2$. Hence, in slot 1, node 1 and all nodes in S_2 are neighborhood optimum. Thus, $A(2) = \{1\} \cup S_2$. All nodes are satisfied under $A(2)$. Hence, MUN terminates at the end of slot 1 after selecting $\{1, P+1, \dots, 2P-3\}$ (weight $P-2 = \frac{(N-2)}{2}$). Thus, MUN selects a dominating set of weight $\frac{N-2}{4} W(D_{\min})$. Greedy and MUNEN select $\{2, P+1\}$ (weight 2).

Corollary 1: MUNEN selects a dominating set whose weight is at most $H(d+1)W(D_{\min})$.

Proof: Corollary 1 follows directly from Theorem 1. Theorem 1 establishes that eventually both Greedy and MUNEN select the same set of nodes. Since Greedy attains an approximation ratio of $H(d+1)$ [2], so does MUNEN. ■

Since $H(d+1)$ is $O(\log(N))$, MUNEN's approximation ratio is $O(\log(N))$.

Note that under MUNEN a node can determine whether it is satisfied under $MD(t)$ only if it knows whether any of its neighbors is in $MD(t)$. Each node knows whether it is in $MD(t)$ since it decides whether to join this set. Thus, a node can determine whether it is extended neighborhood optimum under $MD(t)$ by exchanging messages with its extended neighbors. Thus MUNEN can be executed in a distributed manner.

We now examine whether further simplifications of MUNEN still attain the best attainable approximation ratio of $O(\log(N))$. We first introduce the concept of "neighborhood optimum".

Definition 8: Let $A \subseteq V$. A node u is *neighborhood optimum under A* if for every other node v in N_u (i) either $w_u/US_A(u) < w_v/US_A(v)$ (ii) or $w_u/US_A(u) = w_v/US_A(v)$ and $u < v$.

Intuitively, a node is neighborhood optimum if it is optimum in its 1-hop neighborhood, while it is extended neighborhood optimum if it is optimum in its 2-hop neighborhood. For example, in Fig. 1, consider $A = \{5\}$. Both 1 and 9 are neighborhood optimum under A, while only 9 is extended neighborhood optimum under A.

We consider the natural simplification of MUNEN, MUN in which a node selects itself if and only if it is neighborhood optimum under the current selected set. MUN is otherwise similar to MUNEN. For example, in Fig. 1, before the first iteration, MUN does not select any node, and $1, 5, 9$ are all neighborhood optimum with respect to the empty set. Thus, MUN selects $1, 5, 9$ in the first iteration and terminates in 1 iteration.

Consider a network in which all nodes have unit weight. Figure 3 demonstrates that for any given N, there exists such a network in which MUN selects a dominating set whose weight is at least $\frac{N-2}{4}W(D_{\min})$. Thus, MUN's approximation ratio is $\Omega(N)$. Note that a trivial selection algorithm that selects all nodes has $O(N)$ approximation ratio. Thus, in the worst case, MUN's performance has the same order as that of this trivial selection algorithm.

The trivial selection algorithm can be considered as an algorithm in which a node selects itself if it is "optimum within 0 hops." Thus, the approximation ratio is $\Omega(N)$ when the selection criterion for nodes is optimality within 0 or 1 hop. The approximation ratio reduces to the best possible value of $O(\log(N))$ when the selection criterion for nodes is optimality within 2 hops, and remains the same even when the selection criterion is changed to optimality within p hops where $p > 2$. Thus, the approximation ratio exhibits a threshold phenomenon.

We provide the intuition behind the above threshold phenomenon. For simplicity, assume that all nodes have equal weights. Note that the neighborhoods of 2 nodes that are at a distance of 2 hops may have a large intersection. For example, in Fig. 3, nodes $P + 1, \ldots 2P - 3$ have very similar neighborhoods. Thus, if all nodes that are neighborhood optimum are selected, several of these nodes will cover the same set of unsatisfied neighbors, and thus selecting all neighborhood optimum nodes is redundant. This explains why the approximation ratio is poor when the selection criterion for nodes is optimality within 1 hop. Now, the neighborhoods of the extended neighborhood optimum nodes do not overlap as these nodes are separated by at least 3 hops. Thus, these nodes do not cover the same set of unsatisfied neighbors, and hence selecting extended neighborhood optimum nodes eliminates the redundancy caused due to selection of neighborhood optimum nodes (Fig. 3). For example, in Fig. 3, MUNEN selects nodes $\{2, P+1\}$, whereas MUN selects $\{1, P + 1, \ldots 2P - 3\}$. This explains why the approximation ratio significantly improves when the selection criterion for nodes is optimality within 2 hops. The approximation ratio does not decrease any further with increase in the number of hops, as the overlap in neighborhoods of simultaneously selected nodes can not be reduced any further.

III. Proof of Theorem 1

We first outline the proof. We consider a hybrid algorithm $\mathcal{A}_i$ that selects nodes as per Greedy in the first i slots and then selects nodes as per MUNEN in the remaining slots. Clearly, Greedy and MUNEN are $\mathcal{A}_N$ and $\mathcal{A}_0$ respectively. We prove that for each i, $\mathcal{A}_i$ and $\mathcal{A}_{i+1}$ select the same set of nodes. Thus, MUNEN and Greedy select the same set of nodes.

This section is organized as follows. In subsection III-A, we first define the terminologies which we use throughout the proof. In Subsection III-B, we state and prove a lemma that we use to prove Theorem 1. Finally, we prove Theorem 1 in Subsection III-C.

A. Terminologies

Definition 9: Let $h(u, v)$ be the minimum number of hops between nodes u, v. The minimum distance $d(U, W)$ between the nodes in U and W is $\min_{u\in U, v\in W} h(u, v)$.

Definition 10: Let $A \subseteq V$. LOCAL(A) is the set of the extended neighborhood optimum nodes under A.

Note that if two nodes are extended neighbors of each other, they can not be extended neighborhood optimum under the same set A. Thus, no two nodes in LOCAL(A) can be within two hops of each other.

Definition 11: Let $G_0 = \{v_1, \ldots, v_i\}$ be the set of nodes selected by $\mathcal{A}_i$ at the end of iteration i.

Then, $G_0 \cup D_0$ is the set of nodes selected by $\mathcal{A}_{i+1}$ at the end of iteration $i+1$ where $D_0 = v_{i+1}$. Note that the definitions of time slots in MUNEN and iterations in Greedy are equivalent. Hence, in MUNEN, we refer to "slots" as "iterations".

Definition 12: Let j^{MUNEN} denote the j^{th} iteration of executing MUNEN.

Definition 13: Let Y_j be the set of nodes selected in iteration j^{MUNEN} of $\mathcal{A}_i$ and Z_j be the set of nodes selected in iteration j^{MUNEN} of $\mathcal{A}_{i+1}$ where $j = 1, 2, \ldots$.

Definition 14: Let X_j be the set of nodes selected by $\mathcal{A}_i$ just before iteration j^{MUNEN} and W_j be the set of nodes selected by $\mathcal{A}_{i+1}$ just before iteration j^{MUNEN} where $j = 1, 2, \ldots$

Clearly, $X_1 = G_0$ and $W_1 = G_0 \cup D_0$. Let $W_0 = W_1 \setminus v_{i+1} = G_0$.

By definition, Y_j =LOCAL(X_j) and Z_j =LOCAL(W_j). Thus, no two nodes in Y_j (Z_j) can be within 2 hops of each other.

B. Supporting Lemma

Lemma 1: There exist sets $D_0, D_1, D_2, \ldots$ and $S_0, S_1, \ldots$ such that for all $p = 0, 1, \ldots$

1) $US_{X_{p+1}}(u) = US_{W_p}(u)$, $\forall\ u \in D_p$,
2) $W_{p+1} = X_{p+1} \cup D_p$,
3) $D_p \cap S_p = \phi$, $Y_{p+1} = D_p \cup S_p$,
4) $US_{X_{p+1}}(u) = US_{W_{p+1}}(u)$, $\forall\ u \in S_p$,
5) $X_{p+2} = W_{p+1} \cup S_p$ and
6) $D_{p+1} \cap S_p = \phi$, $Z_{p+1} = S_p \cup D_{p+1}$.

Proof: We prove using induction.

We first consider $p = 0$.

Since $G_0 = X_1 = W_0$, $US_{X_1}(u) = US_{W_0}(u)\ \forall\ u \in D_0$. Thus, (1) follows.

Now, $W_1 = G_0 \cup D_0 = X_1 \cup D_0$. Thus, (2) follows.

We now prove that there exists a set S_0 such that (3), (4) and (5) hold.

Note that Y_1 = LOCAL(X_1). Let us consider the 1^{MUNEN} iteration or the $(i+1)^{th}$ iteration of $\mathcal{A}_i$. Note that $\mathcal{A}_i$ and $\mathcal{A}_{i+1}$ selects the same nodes in the same order in the first i iterations. Therefore, in these iterations, when a node v is selected by $\mathcal{A}_i$, it has the same number of unsatisfied neighbors as when it is selected by $\mathcal{A}_{i+1}$. Since the node v_{i+1} has been selected by Greedy at the $(i+1)^{th}$ iteration of $\mathcal{A}_{i+1}$, this means that v_{i+1} is globally optimum under G_0 and hence it is also extended neighborhood optimum under G_0. Since $X_1 = G_0$, $v_{i+1} \in$ LOCAL(X_1), and hence $D_0 \subseteq$ LOCAL(X_1). Let $S_0 = Y_1 \setminus D_0$. Clearly, (3) holds.

From (3), $D_0 \cup S_0 = Y_1 =$ LOCAL(X_1) and $D_0 \cap S_0 = \phi$. Thus, $d(D_0, S_0) \geq 3$. Thus,

a) $US_{X_1}(u) = US_{X_1 \cup D_0}(u)$, $\forall\ u \in S_0$, and
b) $US_{X_1}(u) \geq US_{X_1 \cup D_0}(u)$, $\forall\ u$.

Now, (4) follows from (2) and (a).

Now, $X_2 = X_1 \cup Y_1$. Now, from (3) and since $X_1 = G_0$, $X_2 = G_0 \cup D_0 \cup S_0$. Since $W_1 = G_0 \cup D_0$, $X_2 = W_1 \cup S_0$. Thus, (5) holds.

We now show that there exists a set D_1 such that (6) holds. Since $D_0 \cup S_0 =$ LOCAL(X_1), each node in S_0 is extended neighborhood optimum under X_1. Now, from (a) and (b), it follows that each node in S_0 is also extended neighborhood optimum under $X_1 \cup D_0$. Hence, $S_0 \subseteq$ LOCAL($X_1 \cup D_0$) = LOCAL(W_1) = Z_1. Let $D_1 = Z_1 \setminus S_0$. Clearly, (6) holds. Thus, the base case holds.

We now assume that there exists sets $D_1, D_2, \ldots, D_k$ and $S_0, S_1, \ldots S_{k-1}$ such that (1) to (6) hold for $p = 0, 1, \ldots, k-1$. We prove that there exists sets D_{k+1}, S_k such that (1) to (6) hold for $p = k$.

From (6) of induction hypothesis, $S_{k-1} \cup D_k = Z_k =$LOCAL(W_k), and $D_k \cap S_{k-1} = \phi$. Thus, $d(S_{k-1}, D_k) \geq 3$. Hence, $US_{W_k \cup S_{k-1}}(u) = US_{W_k}(u)$ for each $u \in D_k$. From (5) of induction hypothesis, $W_k \cup S_{k-1} = X_{k+1}$. Now, (1) follows from the above.

Now, $W_{k+1} = W_k \cup Z_k$. Now, from (6) of induction hypothesis, $W_{k+1} = W_k \cup S_{k-1} \cup D_k$. Next, from (5) of induction hypothesis, $W_{k+1} = X_{k+1} \cup D_k$. Thus, (2) holds.

We now show that there exists a set S_k such that (3), (4) and (5) hold.

Since from (6) of induction hypothesis, $S_{k-1} \cup D_k = Z_k =$LOCAL(W_k), each node in D_k is extended neighborhood optimum under W_k. Since from (5) of induction hypothesis, $W_k \subseteq X_{k+1}$, $US_{X_{k+1}}(u) \leq US_{W_k}(u)\ \forall\ u$. Now, from (1), each node in D_k is extended neighborhood optimum under X_{k+1}. Thus, $D_k \subseteq$LOCAL$(X_{k+1}) = Y_{k+1}$. Let $S_k = Y_{k+1} \backslash D_k$. Clearly, S_k satisfies (3).

From (3), $D_k \cup S_k = Y_{k+1} =$ LOCAL(X_{k+1}), and $D_k \cap S_k = \phi$. Thus, $d(D_k, S_k) \geq 3$. Hence,

i) $US_{X_{k+1}}(u) = US_{X_{k+1} \cup D_k}(u)$, $\forall\ u \in S_k$, and

ii) $US_{X_{k+1}}(u) \geq US_{X_{k+1} \cup D_k}(u)$, $\forall\ u$.

Now, (4) follows from (2) and (i).

Now, $X_{k+2} = X_{k+1} \cup Y_{k+1}$. Thus, from (3), $X_{k+2} = X_{k+1} \cup D_k \cup S_k$. Next, from (2), $X_{k+2} = W_{k+1} \cup S_k$. Thus, (5) holds.

We now show that there exists a set D_{k+1} such that (6) holds. From (3) $D_k \cup S_k = Y_{k+1} =$ LOCAL(X_{k+1}). Thus, each node in S_k is extended neighborhood optimum under X_{k+1}. Now, from (i) and (ii), it follows that each node in S_k is also extended neighborhood optimum under $X_{k+1} \cup D_k$. Hence, $S_k \subseteq$ LOCAL$(X_{k+1} \cup D_k) =$ LOCAL$(W_{k+1}) = Z_{k+1}$. Let $D_{k+1} = Z_{k+1} \setminus S_k$. Clearly, D_{k+1} satisfies (6).

Thus, (1) to (6) hold for all p. ■

C. *Proof of Theorem 1*

We first prove that the set of nodes $MD(N+1)$ is the same as that selected by Greedy when Greedy terminates. Then we prove the convergence guarantee for MUNEN.

Proof: We first prove that $\mathcal{A}_i$ and $\mathcal{A}_{i+1}$ select the same set of nodes. Let $\mathcal{A}_{i+1}$ terminate just before iteration j^{MUNEN}, where $j \geq 1$. Thus, $Z_j = \phi$. Thus, $W_j = W_{j+1}$. Also, from (6) of lemma 1 $S_{j-1} = D_j = \phi$. Thus, from (3) of lemma 1 $Y_j = D_{j-1}$. From (2) of lemma 1, $W_{j+1} = X_{j+1} \cup D_j = X_{j+1}$. Also, $Z_{j+1} = \phi$. Thus, as before, $S_j = \phi$. Thus, $Y_{j+1} = D_j \cup S_j = \phi$. Thus, $\mathcal{A}_i$ terminates at the end of iteration j^{MUNEN}. The result follows since $X_{j+1} = W_{j+1} = W_j$.

If $\mathcal{A}_{i+1}$ terminates before the end of i iterations, since $\mathcal{A}_i$ and $\mathcal{A}_{i+1}$ select nodes in the same manner before the end of i iterations, $\mathcal{A}_i$ and $\mathcal{A}_{i+1}$ terminate at the same iteration with the same number of nodes. Thus, the result follows in this case as well.

Hence, $\mathcal{A}_i$ and $\mathcal{A}_{i+1}$ select the same set of nodes. Thus, $\mathcal{A}_0$ (MUNEN) and $\mathcal{A}_N$ (Greedy) select the same set of nodes. The first part follows.

Now we prove the convergence guarantee for MUNEN. Until all nodes are satisfied, MUNEN selects at least one node in each iteration as the node that is globally optimum under $MD(t)$ is extended neighborhood optimum under $MD(t)$. Now, the result follows since the network has only N nodes. ■

IV. Distributed Algorithms for Approximating Generalizations of Dominating Sets

We consider two important generalizations of the dominating set problem: (i) multi-cover (subsection IV-A) (ii) and k-dominating set (subsection IV-B). Both these problems are NP-hard [2]. Furthermore guaranteeing an approximation ratio of lower than $O(\log(N))$ is also NP-hard in both cases [2]. Greedy can be generalized to attain the best possible approximation ratio of $O(\log(N))$ in both cases, but as expected the generalizations are centralized. Finally, other existing algorithms that attain the approximation ratio of $O(\log(N))$ or any other meaningful performance guarantee are also centralized. We demonstrate that in both cases MUNEN can be generalized to provide a distributed approximation that selects the same nodes as the generalizations of Greedy; the generalizations of MUNEN therefore attain the approximation ratio of $O(\log(N))$ as well.

A. Distributed Approximation For Multi-cover Problem

We first describe the multi-cover problem. The utility of the multi-cover problem is that it allows different coverage requirements for different nodes. Specifically, a node i now needs to be covered by k_i nodes in the selected set, where $k_1, \ldots, k_N$ are integers and k_i can be different for different nodes.

Definition 15: A *multi-cover* M in G is a set of nodes such that every node i in V is covered by k_i nodes in M. A *minimum weighted multi-cover* $M_{\min}$ is a multi-cover that has the minimum weight among all multi-covers.

Refer to Fig. 1 for example of a multi-cover.

It is worthwhile to observe the following. First, depending on the coverage requirements, a multi-cover may not exist in general. For example, if $V = \{1, 2\}$, $E = \{(1, 2)\}$ and $k_1 = 3$, no selection of nodes can cover node 1 with 3 nodes. Thus, a multi-cover does not exist in this case. A dominating set however always exists. Next, the minimum weighted dominating set can be computed as a special case of the minimum weighted multi-cover problem when $k_i = 1$ for each node i.

Consider a subset A of nodes. Given the multi-cover requirement, a node u is *satisfied under* A if it is covered by k_u node in A, and *unsatisfied under* A otherwise.

The only modification required in Greedy and MUNEN is that they must use the above definition for "satisfaction" and "un-satisfaction" under given subsets. Theorem 1 still holds. Corollary 1 can be generalized as follows.

Corollary 2: If there exists at least one multi-cover in the network, MUNEN selects a multi-cover whose weight is at most $H(d + 1)W(M_{\min})$.

Proof: Corollary 2 also follows from Theorem 1 in this case. ■

B. Distributed Approximation For k-dominating Set Problem

We first describe the k-dominating set problem. The utility of the k-dominating set problem is that it allows larger coverage distances. Specifically, each node u now needs to be within k hops of some node in the selected set, where $k \geq 1$.

Definition 16: A *k-dominating set* D^k in G is a set of nodes such that for any node u in V there exists a node v in D^k such that v is within k hops of u in G. A *minimum weighted k-dominating set* $D^k_{\min}$ is a dominating set that has the minimum weight among all k-dominating sets.

Refer to Fig. 1 for example of a k-dominating set.

MUNEN and Greedy can be generalized as follows. Consider a graph $G^k = (V, E^k)$ which can be obtained from G by adding an edge between any two vertices $u, v \in V$ if and only if they are within k hops of each other in G. Note that a set $S \subseteq V$ is a k-dominating set in G if and only if it is a dominating set in G^k. Thus, a k-dominating set in G can be obtained by computing a dominating set in G^k using MUNEN or Greedy. But, every node now needs information about its $2k$-hop neighborhood in G in order to execute MUNEN. Again, MUNEN and Greedy select the same set of nodes and Theorem 1 holds. Corollary 1 also holds with $D_{\min}$ substituted by $D^k_{\min}$ and d being the maximum degree of a node in G^k.

V. Reducing network-wide communication for the maximum coverage problem

The minimum weighted dominating set problem minimizes the weight of the selected set of nodes subject to covering all nodes in the network. The dual "maximum coverage" problem is to maximize the number of covered nodes subject to upper bounding the weight of the selected set of nodes below a pre-specified number k. The dual problem is also NP-hard [2]. Greedy can however be generalized to select a set whose weight is at most k and which covers at least $(1-1/e)$ times the number of nodes covered by the optimal selection. But, this generalization is again centralized and requires $3k$ network-wide broadcasts. We now generalize MUNEN to select the same set of nodes as Greedy while using only a constant (3) broadcasts irrespective of k and the network. Thus, using only limited number of network-wide communications, MUNEN attains the coverage ratio of $1-1/e$, which is also the best possible for all polynomial complexity algorithms unless $P=NP$ [2].

We first define the maximum coverage problem. We initially assume that all nodes have unit weight.

Definition 17: The *coverage set* $C(S)$ of a set S of nodes is the union of the neighborhoods of nodes in S. Thus, $C(S) = \cup_{u \in C} N_u$. The *coverage number* of a set S is the cardinality of the coverage set $C(S)$. The *maximum coverage* problem is to select a set S_{opt} whose cardinality is at most k and which has the largest coverage number among all sets of cardinality k or less.

Refer to Figure 1 for an example of S_{opt} for $k=2$.

Definition 18: The *coverage ratio* of an algorithm for the maximum coverage problem is the worst (minimum) value of the ratio between the coverage number of the set of nodes selected by the algorithm and $|C(S_{\text{opt}})|$ in any network.

As stated before, Greedy can be generalized to attain a coverage ratio of $1-1/e$ [2]. The generalization, referred to as G-Greedy, is simply to terminate Greedy at the end of k iterations, and use the set $GD(k+1)$ selected by Greedy as an approximate solution of the maximum coverage problem. For example, in Fig. 1, if $k=1$, G-Greedy selects node 5. In each iteration, Greedy needs to determine the node that has the maximum number of unsatisfied neighbors. Thus, G-Greedy requires $O(k)$ network-wide communications.

We now generalize MUNEN so as to select the same set of nodes as Greedy while using a constant number of network-wide communications. We first describe the following properties of MUNEN used in the generalization.

Lemma 2: For each t, $GD(t) \subseteq MD(t)$.

Thus, at the end of k slots, MUNEN selects the nodes selected by G-Greedy, but may select additional nodes. Recall that MUNEN does not require network-wide communication in any slot, but identifying these additional nodes requires a constant number of network-wide communications. We first introduce the following notion that allows the identification of these nodes.

Definition 19: Let a node u be selected by MUNEN in slot $t_u \le k$. A node u has higher priority than another node v if either (i) $US_{MD(t_u)}(u) > US_{MD(t_v)}(v)$ or (ii) $US_{MD(t_u)}(u) = US_{MD(t_v)}(v)$ and $u < v$.

For example, in Fig. 1, let $k=1$. MUNEN selects 5 and 9 in the first slot. Thus, $t_5 = t_9 = 1$, $US_{MD(t_5)}(5) = 6, US_{MD(t_9)}(9) = 5$. Thus, 5 has higher priority than 9. Clearly, under this definition, either u has higher priority than v or v has higher priority than u.

Lemma 3: The nodes in $GD(k+1)$ constitute the nodes with k largest priorities in $MD(k+1)$.

We prove Lemmas 2 and 3 in the appendix.

We now describe the generalization of MUNEN, which we refer to as G-MUNEN. At the end of k slots, nodes stop executing the selection strategy in Fig. 2 and participate in network-wide communications to determine k nodes with largest priorities amongst those that are already selected. These nodes constitute the approximate solution for the maximum coverage problem obtained by G-MUNEN.

For example, in Fig. 1, let $k = 1$. G-MUNEN executes MUNEN in the first slot and selects 5 and 9. As discussed above, 5 has higher priority than 9. Since $k = 1$, G-MUNEN retains the node with the highest priority, 5.

G-MUNEN has the following analytical guarantees.

Theorem 2: G-Greedy and G-MUNEN select the same nodes.

Theorem 2 follows from Lemmas 2 and 3.

Corollary 3: MUNEN has coverage ratio $1 - 1/e$.

Corollary 3 follows from Theorem 2 and the coverage ratio of Greedy [2].

We now describe how G-MUNEN can discover k nodes with largest priorities using 3 broadcasts. Some pre-determined root node broadcasts a query packet along a spanning tree. The query packet has k tuples, and initially each tuple has value $(0, 0)$. Each node u includes its identity u and $US_{MD(t_u)}(u)$ if its priority is higher than that of the least priority node v in the list, and removes the entry corresponding to v. Leaf nodes of the spanning tree return the query packet towards the root which re-broadcasts the query packet. Thus, every node knows which k nodes have the largest priorities. Note that in each iteration, G-Greedy can similarly learn the node that has the maximum number of unsatisfied neighbors using 3 broadcasts, and therefore requires $3k$ broadcasts overall.

We now mention important distinctions in the case in which nodes have unequal weights. First, the coverage number should now be the weight of the coverage set. We also need to select the set of nodes whose weight (and not cardinality) is less than or equal to k. G-Greedy should now terminate in the iteration l in which either all nodes are covered or $W(GD(l)) \leq k < W(GD(l+1))$. Thus, in the worst case, G-Greedy requires $3N$ broadcasts. For G-MUNEN, the priority of a node u is $US_{MD(t_u)}(u)/w_u$. Nodes execute MUNEN for N slots. Subsequently, m highest priority nodes are retained such that their total weight is less than or equal to k, and the total weight of $m + 1$ highest priority nodes exceed k. These nodes are identified using 3 broadcasts. Thus, G-MUNEN may require larger number of iterations as compared to G-Greedy, but still requires constant number of network-wide communications.

VI. Conclusion, Discussion and Related Work

The problem of approximating the node domination and coverage problems has been well investigated [1], [8]. However, to the best of our knowledge, all the existing approximation algorithms that provide the worst case performance guarantees are centralized [1], [8]. Performance guarantees have been obtained in the expected case using distributed randomized algorithms [3], [4]. In addition, several interesting heuristics have been proposed whose performance have been verified using simulations [7]. Several large scale networks however require both distributed selection strategies and the worst case performance guarantees. Our contribution has been to prove that a simple distributed approximation algorithm, MUNEN, emulates the best known centralized approximation algorithm with worst case performance guarantees in every network for a large class of these problems. We further demonstrate that the approximation ratio exhibits a threshold phenomenon with respect to the amount of information used in the decision process.

The work that is most relevant to ours is that by Liang *et. al.* [5]. They have proposed an algorithm that is similar to MUNEN. They have also observed one of the results we prove, i.e., MUNEN emulates the best known centralized algorithm for the k-dominating set problem. Their proof is however flawed, and is essentially based on intuitive arguments. For example, they ignore the key fact that the two algorithms select different sets of nodes in each slot, and the selection strategies of the two may therefore be based on different number of uncovered neighbors of the nodes. Note that the result that MUNEN selects the same set of nodes as a centralized approximation algorithm in spite of using only local information is both strong and counter-intuitive. For example, this result allowed us to (i) generalize MUNEN to approximate using local information a large class of domination and coverage problems, and (ii) demonstrate the interesting threshold phenomenon. It is therefore imperative that this result be proved correctly. We prove this result using the concept of a hybrid algorithm, which use a centralized decision process for certain number of iterations and MUNEN for the rest. We prove that the set of nodes selected by the hybrid algorithm does not depend on the number of iterations it uses for each strategy. We consider both the result and the proof technique as the major contribution of this paper. We believe that given MUNEN's simplicity, distributed nature of operations, and its strong analytical performance guarantee, it is likely to be widely used in resource constrained large scale networks.

Finally, recently, several coverage problems have been considered in context of sensor networks. These coverage problems select nodes so as to cover an area under consideration, while we select nodes so as to cover other nodes [6], [9]. Thus the sensor-coverage problems investigate the impact of several node deployment strategies, while we investigate the impact of selection strategies once the nodes are deployed and the topology is given.

APPENDIX
PROOF OF LEMMAS 2 AND 3

In Subsection A, we first define the terminologies which we use throughout the proof. Then, in Subsection B, we state and prove supporting lemmas, Lemmas 4 to 6. Finally, in Subsection C we prove Lemmas 2 and 3.

A. Terminologies

Definition 20: Let $A \subseteq V$. The span of node v, $\text{span}_A(v)$, is the tuple $(US_A(v), v)$.

We define an ordering on the spans. Let $\text{span}_A(v) = (v_1, v_2)$ and $\text{span}_W(u) = (u_1, u_2)$. Now, $\text{span}_A(v) < \text{span}_W(u)$ if (i) $v_1 < u_1$ or (ii) $v_1 = u_1$ and $v_2 < u_2$.

Clearly, $\text{span}_A(v) \neq \text{span}_A(u)$ for any A, u, v.

Definition 21: Let $I = \{v_1, v_2, v_3, \ldots, v_l\}$ be the set of nodes selected by Greedy and hence by MUNEN.

Definition 22: Let $v \in I$. The *span-select-MUNEN* of v is $\text{span}_{MD(j)}(v)$ if MUNEN selects v in the j^{th} iteration. The *span-select-Greedy* of v is $\text{span}_{GD(j)}(v)$ if Greedy selects v in the j^{th} iteration.

Note that node u has higher priority than v if it has higher *span-select-MUNEN* than v.

Definition 23: Let T_j be the set of nodes selected in the j^{th} iteration of MUNEN e.g $T_j = MD(j+1) \setminus MD(j)$.

B. Supporting Lemmas

Lemma 4: If node v is in I, its *span-select-MUNEN* and *span-select-Greedy* are identical.

Proof: Let $v \in I$. From Theorem 1, for any $i, j \in \{0, \ldots, N\}$, both $\mathcal{A}_i$ and $\mathcal{A}_j$ select v. Let P_i be the set of nodes $\mathcal{A}_i$ selects just before $\mathcal{A}_i$ selects v. We will prove that $US_{P_i}(v) = US_{P_j}(v)$. Thus, $US_{P_0}(v) = US_{P_N}(v)$. The lemma now follows.

We now prove that $US_{P_i}(v) = US_{P_j}(v)$ for any $v \in I$ and $i, j \in \{0, \ldots, N\}$. Clearly, it is sufficient to prove the result for i, j such that $i < N$ and $j = i + 1$. Let $\mathcal{A}_i$ select v in iteration l. Let $l \leq i$. Since $\mathcal{A}_i$ and $\mathcal{A}_{i+1}$ select the same nodes in each iteration $1, \ldots, i$, the result holds. Now, let $l > i$, i.e., $v \in Y_m$ for some m, where $m \geq 1$. Then $P_i = X_m$. From (3) of lemma 1, $v \in D_{m-1}$ or $v \in S_{m-1}$. Let $v \in S_{m-1}$. From (6) of lemma 1, $v \in Z_m$. Then $P_{i+1} = W_m$. The result follows from (4) of lemma 1. Now, let $v \in D_{m-1}$. Let $m > 1$. From (6) of lemma 1, $v \in Z_{m-1}$. Then $P_{i+1} = W_{m-1}$. The result follows from (1) of lemma 1. Now, let $m = 1$, i.e., $v \in D_0$. Thus, $v = v_{i+1}$ and $\mathcal{A}_{i+1}$ selects v in iteration $i + 1$. Thus, $P_{i+1} = G_0 = W_0$. The result follows from (1) of lemma 1. ∎

Lemma 5: In MUNEN, in any iteration j, $\forall\ u$, $\text{span}_{MD(j)}(u) \geq \text{span}_{MD(j+1)}(u)$

Proof: First assume that MUNEN selects u in the j^{th} iteration or before. Then $\text{span}_{MD(j+1)}(u) = (0, u)$. Thus, $\text{span}_{MD(j+1)}(u) \leq \text{span}_{MD(j)}(u)$. Now let MUNEN not select u in the j^{th} iteration or before (MUNEN may select u after the j^{th} iteration or may never select u). Let v be the node nearest to u that MUNEN selects in the j^{th} iteration. If $h(u, v) > 2$, then $\text{span}_{MD(j)}(u) = \text{span}_{MD(j+1)}(u)$. If $h(u, v) \leq 2$, then $\text{span}_{MD(j)}(u) \geq \text{span}_{MD(j+1)}(u)$. ∎

Lemma 6: For any iteration j of MUNEN, $\exists\ v \in T_j$ such that $\text{span}_{MD(j)}(v) > \text{span}_{MD(j+1)}(u)\ \forall\ u \in T_{j+1}$.

Proof: Let there not exist $v \in T_j$ such that $\text{span}_{MD(j)}(v) > \text{span}_{MD(j+1)}(u)\ \forall\ u \in T_{j+1}$. Now, $\text{span}_{MD(j)}(v) \neq \text{span}_{MD(j+1)}(u)$ for all $v \in T_j$ and $u \in T_{j+1}$, since $v \neq u$ if $v \in T_j$ and $u \in T_{j+1}$. Therefore, for each node $v \in T_j$, there must exist some node $u \in T_{j+1}$ such that $\text{span}_{MD(j+1)}(u) > \text{span}_{MD(j)}(v)$. Thus, there must exist at least one node $u_0 \in T_{j+1}$ such that $\text{span}_{MD(j+1)}(u_0) > \text{span}_{MD(j)}(v)\ \forall\ v \in T_j$. From lemma 5, $\text{span}_{MD(j)}(u_0) \geq \text{span}_{MD(j+1)}(u_0)$. Thus, $\text{span}_{MD(j)}(u_0) > \text{span}_{MD(j)}(v)\ \forall\ v \in T_j$.

Since $u_0 \in T_{j+1}$, MUNEN does not select u_0 in the j^{th} iteration. Thus, there exists a node u_1 s.t. $h(u_0, u_1) \leq 2$ and $\text{span}_{MD(j)}(u_0) < \text{span}_{MD(j)}(u_1)$. Again, either MUNEN selects u_1 in the j^{th} iteration, or there exists a node u_2 s.t. $h(u_1, u_2) \leq 2$ and $\text{span}_{MD(j)}(u_1) < \text{span}_{MD(j)}(u_2)$. Since the number of nodes is finite, continuing in this manner, we obtain a node u_p such that $\text{span}_{MD(j)}(u_0) < \text{span}_{MD(j)}(u_p)$ and $u_p \in T_j$. But, this contradicts the fact that $\text{span}_{MD(j)}(u_0) > \text{span}_{MD(j)}(v)\ \forall\ v \in T_j$. ∎

C. *Proof for Lemmas 2 and 3*

Proof for Lemma 3: It is sufficient to show that nodes in $GD(k+1)$ constitute the nodes with k largest *span-select-MUNEN* in $MD(k+1)$. From the selection strategy in Greedy, the nodes in $GD(k+1)$ constitute the nodes with k highest *span-select-Greedy* in I. Hence, from lemma 4, the nodes in $GD(k+1)$ constitute the nodes with k highest *span-select-MUNEN* in I. Since different nodes selected by MUNEN have different *span-select-MUNEN*, it is now sufficient to prove that the nodes with k highest *span-select-MUNEN* in $MD(k+1)$ are those with k highest *span-select-MUNEN* in I. First, $MD(k+1) \subseteq MD(N+1) = I$. Now, since $k \leq N$, from lemma 6, we know that there exists at least k nodes in $MD(k+1)$ that have larger *span-select-MUNEN* than $u\ \forall\ u \in I \setminus MD(k+1)$. The result follows.

Proof for Lemma 2: Lemma 2 follows directly from lemma 3.

References

[1] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Fundamentals of Domination in graphs*. Marcel Dekker Inc., 1998.

[2] D. Hochbaum. *Approximation Algorithms for NP-Hard Problems*. Thomson Learning, 1996.

[3] L. Jia, R. Rajaraman, and R. Suel. An efficient distributed algorithm for constructing small dominating sets. In *In Proceedings of the 20th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 33–42, 2001.

[4] F. Kuhn and R. Wattenhofer. Constant-time distributed dominating set approximation. In *In Proceedings of the 22nd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 25–32, 2003.

[5] B. Liang and Z. J. Haas. Virtual backbone generation and maintenance for ad hoc network mobility management. In *In Proceedings of the 19th IEEE INFOCOM*, March 2000.

[6] S. Shakkottai, R. Srikant, and N. Shroff. Unreliable sensor grids: Coverage, connectivity and diameter. In *IEEE Infocom*, 2003.

[7] R. Sivakumar, P. Sinha, and V. Bharghavan. Cedar: A core-extraction distributed ad hoc routing algorithm. In *IEEE Journal on Selected Areas in Communications*, volume 17, Aug. 1999.
[8] V. Vazirani. *Approximation Algorithms*. Springer-Verlag, 2001.
[9] H. Zhang and J. Hou. On deriving the upper bound of alpha-lifetime for large sensor networks. *Proceedings of ACM MobiHoc*, 2004.