DISTRIBUTED ONE-HOP ALGORITHM BASED ON STEINER CONNECTED DOMINATING SET IN WIRELESS NETWORKS

Rashid Bin Muhammad
Department of Computer Science
Kent State University
Kent, OH, USA
Email: rmuhamma@cs.kent.edu

ABSTRACT

Multicast routing problem is to find a tree rooted at a source node to all multicast destination nodes. In dominating set problem, we are required to find a minimum size subset of vertices that each vertex is either in the dominating set or adjacent to some node in the dominating set. In this paper, we concentrate on the related problem of finding a connected dominating set of minimum size in which the graph induced by nodes in the dominating set are required to be connected. Instead of conventional spanning tree model, we used Steiner tree in unit-disk graph to connect nodes. The main contributions of this work are fully distributed Steiner tree with approximation ratio of 2 and fully distributed algorithm for multicast backbone structure with an approximation ratio at most 10. The main advantage of the algorithm is that it simplifies the routing process to one in a smaller subnetwork.

KEY WORDS
Virtual backbone, Connected dominating set, Steiner trees, Approximation algorithm, Distributed algorithm.

1 Introduction

Ad hoc wireless network [1, 2] is featured by dynamic infrastructure-less topology, multi-hop communication, limited resources (bandwidth, CPU, battery, etc.) and limited security. It consists of many small and inexpensive sensor nodes that are distributed over a large field to obtain data. These nodes are capable of storing, processing, and transferring data. Due to these characteristics, ad hoc wireless networks are adopted in many military and civil applications such as decision making in the battlefields, search-and-rescue operations in emergencies, data acquisition operations in inhospitable terrain, etc. Furthermore, these characteristics put special challenges in routing protocol design. An essential problem concerning ad hoc wireless networks is to design routing protocols allowing for communication between the host nodes [2]. The dynamic nature of ad hoc networks makes this problem especially challenging. However, in some cases the problem of computing an acceptable virtual backbone can be reduced to the well-known minimum connected dominating set, MCDS, problem in unit-disk graph [3]. The backbone (spin) infrastructure plays an important role in wireless networks for routing, connectivity management, and broadcasting. Although wireless ad hoc networks have no physical backbone infrastructure, nodes in a connected dominating set of the corresponding unit-disk graph can form a virtual backbone. The objective of this paper is to construct the virtual backbone infrastructure with minimal number of multicast nodes using minimal Steiner tree.

A network can be represented as a simple undirected graph $G = (V, E)$ with the set of vertices $V$ representing the set of machines and the set of edges $E$ representing the set of communication links among machines. The set of application nodes $M$ may be present at a subset $V_M$ of nodes in $G$, $V_M$ is typically referred to as a multicast group or cluster. Because the application nodes need to communicate with one another, the network must provide connectivity between the set $V_M$ nodes. Multicasting is a common communication mode in applications wherein nodes in $V_M$ may have to send messages to all other nodes in $V_M$. For example, in a teleconferencing application, audio data from each participant may have to be multicast to all other participants.

In ad hoc networks, most changes occur within a small area of the network. Therefore, we abstract the network structure such that local changes need not be seen globally. This is done by using logical substructure called cluster or clique, which is a complete subgraph. The process of defining these substructures within a complete network topology is called clustering. Some nodes in the cluster may be designated to oversee message routing and channel allocation. These special nodes are called cluster-heads. From a network perspective, each of these clusters, represented by its cluster-head, is a vertex in a graph $G$ and virtual connections between clusters are edges.

In this paper, we addresses the Steiner connected dominating problem. In this problem we are required to dominate only a specific subset $V_M$ of the vertices. The cost of the solution is the size of the smallest connected dominating set that dominates the vertices in $V_M$.


Our method for clustering is based on domination in graph. Chen and Stojmenovic [9] provided a detailed review of these techniques. Wu and Li [10] proposed clustering algorithm based on finding connected dominating sets in a graph. By constraining the dominating set, we can ensure that cluster-heads are adjacent to each other or at least reasonably close to each other. A connected dominated set of a given graph is a dominating set whose induced sub-graph is connected. Unfortunately, finding a minimum size connected dominating set is also NP-complete [11]. Guha and Khuller [12] proposed two approximation algorithms for finding small-connected dominating sets. Das and Barghavan [13] implemented distributed versions of the above two algorithms.

Guha and Khuller proposed the generalize version of the minimum connected domination problem in [12], namely the Steiner connected dominating set problem. They presented two centralized polynomial algorithms that achieve approximation factors of $2H(\Delta)+2$ and $H(\Delta)+2$, where $\Delta$ is the maximum degree and $H$ is the harmonic function.

The general outline of the paper is as follows. Some basic definitions and problem definition are given in Section 2. Section 3 presents distributed Steiner tree approximation; analyze its message and time complexity; and established its correctness. Section 4 deals with one-hop algorithm and its performance. Section 5 presents the distributed implementation of Steiner connected dominating set. Finally, some concluding remarks are collected in Section 6.

## 2 Preliminaries and Problem Definition

The proposed distributed algorithm for Steiner connected dominating set satisfies the following criteria. Firstly, the execution of the algorithm is restricted to multicast nodes. Secondly, each node operates based on its local information; therefore, there is no need for the central node, to keep the information for entire network.

The Steiner problem in networks can be formulated as follows. Given network $G = (V, E)$ and a non-empty set of multicast nodes $V_M \subseteq V$, the problem is to find a subnetwork $T_G(V_M)$ such that there is a path between every pair of nodes and total length $|T_G(V_M)| = \sum_{i \in T_G(V_M)} c(e_i)$ is minimized. Nodes that end up in $T_G(V_M)$ are called Steiner vertices.

A dominating set, $DS$, is a set $D \subseteq V$ such that each vertex in $V \setminus D$ is adjacent to at least one vertex in $D$. If the graph is connected, a connected dominating set, is a dominating set, which is also connected subgraph of $G$.

Among all connected dominating sets of graph $G$, the one with minimum cardinality is called a minimum connected dominating set. Note that the problem of finding an minimum connected dominating set in a graph is equivalent to the problem of finding a spanning tree, with maximum number of leaves. All non-leaf nodes in the spanning tree form the minimum connected dominating set. Note that a maximal independent set, $MIS$, is also a dominating set. It is known that both connected dominating set and minimum connected dominating set problem are $NP$-hard [11]. This remains the case even when they are restricted to planar, unit disk graph [14].

For a graph $G$, if $e = (u, v) \in E$ if and only if length $(e) \leq 1$, then $G$ is called a unit-disk graph. From now on, when we say a “graph $G$”, we mean a “unit-disk graph $G$”. Ad hoc networks can be modeled using unit-disk graphs as follows. The hosts in a wireless network are represented by vertices in the corresponding unit-disk graph, where the unit distance corresponds to the transmission range of wireless devices.

For a minimization problem $P$, the performance ratio of an approximation algorithm $A$ is defined as $\rho_A = \sup_{i \in I} \frac{A_i(\text{opt}_i)}{\text{opt}_i}$, where $I$ is the set of instances of $P$, $A_i$ is the output from algorithm $A$ for instance $i$ and $\text{opt}_i$ is the optimal solution for instance $i$. In other words, $p$ is the supreme of $A_i(\text{opt})$ among all instances of $P$. For Steiner problem in networks, the performance ratio of an approximation algorithm is given by the maximum ratio of then length of the returned Steiner tree and the length of the returned Steiner tree and the length of a Steiner minimum tree, where the maximum is taken over all admissible instances of Steiner problems in networks.

The one-hop open neighborhood $N(v)$ of the node $v$ consists of the set of nodes adjacent to $u$, that is, $N(v) = \{w \in V: uw \in E\}$, and the one-hop closed neighborhood of $v$ is $N[v] = \{w \in V : vw \in E\}$. For a set $S \subseteq V$, the one-hop open neighborhood $N[S]$ is defined to be $\cup_{v \in S} N(v)$ and the one-hop closed neighborhood of $S$ in $N[S] = N(s) \cup S$. For a set of multicast nodes $V_M \subseteq V$, the one-hop open neighborhood $N[V_M]$ is defined to be $\cup_{v \in V_M} N(v)$ and the one-hop closed neighborhood of $V_M$ is $N[V_M] = N(V_M) \cup V_M$.

This paper investigates the problem of constructing a virtual multicast infrastructure with minimum multicast nodes with the fastest forwarding nodes, which are responsible for forwarding multicast packets and reduce redundant broadcast. We accomplished this task by computing the maximal independent set of multicast nodes and connecting them by Steiner tree. For this, we have proposed the fully distributed Steiner approximation. We shall formulate this problem in a unit-disk graph as follows. Given a network $G = (V, E)$ and a subset of $V_M$ of multicast nodes, the problem is to find the smallest subset $F$ of forwarding nodes that induce a connected subgraph and each vertex in $V_M \setminus F$ is one-hop neighbor of at least one vertex in $F$. 

226
3 Distributed Steiner Tree Approximation

This section presents the fully distributed Steiner tree approximation. Multicast routing is to find a routing tree which spans all destination nodes and whose network cost is minimum. A shortest path from a tree $T$ to a non-tree node $v \notin T$ is defined as a shortest path from a tree node $u \in T$ to $v$ and for any node $k \in T$, $u$ satisﬁes $\sum_{i \in sp(u,v)} c(l) \leq \sum_{i \in sp(k,v)} c(l)$, where $sp(u,v)$ denotes the shortest path from $u$ to $v$ in $T$, and $u \in T$ is said to be the node closest to $v \notin T$. The cost from a tree $T$ to a $v \notin T$, $c(T, v)$, deﬁned as $c(T, v) = \sum_{i \in sp(u,v)} c(l)$. A node $w \notin T$ closest to a tree $T$ satisfy $\forall \forall : c(T, w) \leq (T, v)$. Clearly, these deﬁnitions also hold for paths, which are a special case of trees, single-branch trees.

The idea of the approximation to construct the routing tree is based on Prims MST algorithm. Start constructing a routing tree from the source node. Select any node not in the routing tree and the shortest path between the tree node and not in the tree, respectively. Now we formalize the above idea in the following Steiner tree approximation as follows.

**Steiner Tree Approximation (T, W:links(V))**

while $W \neq \phi$
do for $u \in T$
do $m(u) \leftarrow 0$
$Q_0 \leftarrow \phi; Q_1 \leftarrow T; u \in T$
while $Q_1 \neq \phi \wedge u \in W$
do $v \in \{w \in Q_1 : l(w) = \min_{w \in Q_1} m(w)\}$
if $u \notin w$ then
for $u' \in w$ then
if $u' \notin Q_1$ then
$m(u') \leftarrow \min\{l(u), l(u) + d(u, u')\}$
$Q_0 \leftarrow Q_0 \cup \{u\}; Q_1 = Q_1 \setminus \{u\}$
end if
end for
end if
end do
end do
end do
\end{verbatim}

**Proof.** Suppose a cycle is formed when adding the edge to $u$ and $v$. Here, $v$ is the closest vertices to $u$ and path $<uv>$ is the shortest path. To form the cycle, $<uv>$ must meet another tree node, because $<uw>$ is the shortest path and itself does not contain a cycle. Suppose $<uwk>$ meets tree node $v'$, where $v \neq v'$. It is easy to see that $v'$ is a tree node which is closer to $u$ than $v$, a contradiction. Thus, lemma holds.

**Cost Analysis**

To evaluate the distributed Steiner tree approximation, we use the two important criteria namely, the number of messages and the time needed to construct the Steiner tree.

**Lemma 2.** In the worst-case, Steiner tree approximation uses $O(2|V|)$ number of messages and run in $O(|V| - 1)|E|log|V|$ time.

**Proof.** Every node in the network system executes the same Steiner tree approximation. In addition, we count the transmission of a message from a source to a destination as one message. Our approximation uses the following message to construct a tree. It needs $m$ number of messages from source node to the $m$ destination nodes and at most $|V| - 1$ messages from $|V| - 1$ destinations to inform source of the completion of the routing tree. Therefore, it takes $2|V|$ number of messages to construct a routing tree. Furthermore, the time approximation finds the shortest path is $O(|E|log|V|)$ provided the set $Q$ is represented by heap data structure (see second while loop in the approximation). First while loop runs for $|V| - 1$ node (to deliver $|V| - 1$ messages). Therefore, total running time of the Steiner tree approximation, in the worst-case, is $O(|V| - 1)|E|log|V|$.

**Lemma 3.** Let $\sum_{e \in T_G(V_M)} c(e_i)$, where $c(e_i)$ is an edge length function $c : E \rightarrow R$, be a total length of a $T_G(V_M)$ produced by Steiner tree approximation. Then every minimum spanning tree MST in the complete network $k$ satisfies $\text{length}(\text{MST}) \leq (2 - \frac{2}{|V_M|}) T_G(V_M)$ where $|V_M|$ is the size of multicast node set.

**Proof.** Let $T_G(V_M)$ be a tree provide by Steiner tree approximation. Embed the $T_G(V_M)$ in the Euclidean plane. Now consider a walk $W$ along the edges of $T_G(V_M)$. This walk visits every terminal exactly once and traverses every edge twice. Therefore, its length is exactly twice the length of $T_G(V_M)$. Let $t$ be the number of leaves in $T_G(V_M)$. Then $W$ consists of $t \leq k$ paths between successive leaves in $T_G(V_M)$. Remove the longest of these paths from $W$. The length of the remaining walk $W'$ is at most $(1 - \frac{1}{t})$ times of $W$. Now notice that following the $W'$ one can construct a spanning tree. Hence, Steiner tree approximation returns a solution whose value is at most twice “far off” with respect to $T$. 

227
4 One Hop Algorithm

In this section, we propose a connected dominating set approximation using minimum Steiner tree approximation for unit-disk graph approximation ratio with \( p \leq 10 \). Since it is efficient to transmit data over a multicast tree therefore, the connectivity in \( V_M \) is realized in terms of a subgraph \( T_{G,M}(V^\prime_M) = (V^\prime_M, E_M) \), where \( T_{G,M}(V^\prime_M) \) is a tree with \( V_M \subseteq V^\prime_M \) and \( E_M \subseteq E \). Therefore, \( T_{G,M}(V^\prime_M) \) forms a multicast spanning tree for the set \( V_M \) but since we are using a Steiner tree approach, \( T_{G,M}(V^\prime_M) \) produced by one-hop algorithm may have nodes addition to those in \( V_M \).

The basic idea of the approximation algorithm is as follows. First, compute a maximal independent set of multicast group \( V_M \) that dominate all multicast nodes in \( G \). Secondly, connect these multicast nodes by using Steiner minimal tree approximation. The one-hop algorithm as follows.

One Hop Algorithm

Step 1. Compute a maximal independent set \( MIS \) for multicast node in graph \( G = (V_M, E_M) \).

Step 2. Use Steiner tree approximation to construct a Steiner tree in \( MIS \). The output of the algorithm is the set of node in \( T_{G,M}(V^\prime_M) = (V^\prime_M, E_M) \).

Note that \( MIS \) is also a dominating set of \( G = (V_M, E_M) \). All \( v \in V_M \) are dominated by the set \( MIS \). The second step constructs the tree \( T_{G,M}(V^\prime_M) = (V^\prime_M, E_M) \) that spans all nodes of \( MIS \) and dominates nodes \( v \in V_M \). Therefore, the nodes of \( T_{G,M}(V^\prime_M) \) form a Steiner connected dominating set of \( V_M \).

Performance Analysis

The performance of one-hop algorithm is based on properties of independent set. The following lemma establishes the relation between the size of the independent set and the size of the connected dominating set in unit-disk graph [15].

Lemma 4. The size of any independent set in a unit-disk graph at most \( 4 \times |D| + 1 \), where \( D \) is a connecting dominating set.

Let \( T_{G,M}(V^\prime_M) \) be the Steiner connected dominating set of multicast nodes in \( G(V_M) \) where \( V_M \subseteq V^\prime_M \). The size of \( MIS \) in \( G \) is bounded by the size of \( T_{G,M}(V^\prime_M) \) using the above lemma, we established the following.

Lemma 5. The size of \( MIS \) in \( G = (V_M, E_M) \) is at most \( 4 \times |T_{G,M}(V^\prime_M)| + 1 \), where \( |T_{G,M}(V^\prime_M)| \) is the size of optimal Steiner connected dominating set.

Proof. Given the graph \( G = (V_M, T_{G,M}(V^\prime_M)) \) and its maximal independent set \( MIS \). \( T_{G,M}(V^\prime_M) \) dominates all nodes in \( V_M \). Furthermore, \( T_{G,M}(V^\prime_M) \) is a connected dominating set of the graph \( G \). Therefore, from Lemma 4, the size of \( MIS \) is at most \( 4 \times |T_{G,M}(V_M)| + 1 \). Therefore, the approximation ratio of one-hop algorithm is at most 10.

Lemma 6. The total number of nodes in the dominating tree produced by one-hop algorithm is at most \( 10 \times |T_{G,M}(V^\prime_M)| + 1 \), where \( V_M \subseteq V \).

Proof. Since \( T_{G,M}(V^\prime_M) \) is a connected dominated set and each multicast node in maximum independent set is dominated by \( T_{G,M}(V^\prime_M) \) therefore, the subgraph of \( MIS \) and \( T_{G,M}(V^\prime_M) \) i.e. \( G(MIS \cup T_{G,M}(V^\prime_M)) \), is a connected graph and has at most \( (|MIS| + |T_{G,M}(V^\prime_M)|) \) number of nodes.

Let \( T^\prime \) be the spanning tree of \( G(MIS \cup T_{G,M}(V^\prime_M)) \). Clearly, \( T^\prime \) is a Steiner tree in \( G \) with at most \( (|MIS| + |T_{G,M}(V^\prime_M)| - 1) \) number of edges. Therefore, the minimum Steiner tree of \( MIS \) must have at most \( (|MIS| + |T_{G,M}(V^\prime_M)| - 1) \) edges.

After the computation of maximal independent set, we execute Steiner tree approximation. Steiner tree \( T \) in a graph having unit weight has at most \( p \times (|MIS| + |T_{G,M}(V^\prime_M)| - 1) \), where \( p \) is at most 2 from above, number of edges. Therefore, the total number of nodes in Steiner tree is at most \( 2 \times (|MIS| + |T_{G,M}(V^\prime_M)| - 1) + 1 \)

\( \leq 10 \times |T_{G,M}(V^\prime_M)| + 1 \)

To implement one-hop algorithm, we used modified version of the algorithm in [16] to compute an \( MIS \) among multicast clusters. The modified algorithm runs in \( O(|V|) \) time. The second step uses the distributed Steiner tree with message complexity, \( O(2|V|) \) and time complexity of \( O(|V| - 1)|E| \log |V|) \). Therefore, the message and time complexity of one-hop algorithm are \( 2|V| \) and \( O(|V| - 1)|E| \log |V| \) respectively.

5 Distributed Implementation

The one-hop algorithm runs by all nodes in the network system. We make the following operational assumptions. First, message send by any node is correctly received by its one-hop neighbor. Second, network topology remains the same during the execution of the algorithm. Finally, each node knows its one-hop neighbor, \( N[V_M] \) and IDs of its one-hop multicast neighbors, \( ID_M \).

The distributed implementation is divided into two phases. First phase constructs the one-hop multicast clusters and identify the cluster-heads and the second phase constructs the tree by distributed Steiner tree approximation to connect these clusters-heads.

In the first phase, we first compute a one-hop \( MIS \) of multicast group by exchanging messages with neighbors using modification of algorithm in [17], the multicast node with lowest \( ID \), \( ID_M \), among neighboring nodes is chosen to be head-node in \( MIS \). Based on the information obtained during message exchanges, each head of the group selects some multicast nodes to be cluster dominators, which dominate all multicast nodes in the cluster. The cluster-head unicasts messages to notify all cluster-dominator along the shortest path. All nodes on the shortest path become member of virtual multicast backbone, VMB.
which are used to connect the cluster-heads and the dominants of group-heads. This phase uses three types of messages. A message broadcast by the multicast node to its neighbors cluster dominator sends messages to other cluster heads. Messages sent by group dominators and dominate to tell their status. Each node maintains a list Mlist, which records next hop information and destination ID_M.

The distributed implementation of the first phase is as follows.

1. Initially, each u ∈ V_M broadcast a message, Msg(sender, dest, list), to its neighbors.
2. Upon receiving the message, node, v, updates its list. If destination field does not match its ID, node v passes the message to its neighbors.
3. Upon receiving the message, multicast node, v ∈ V_M, updates its lists. If v’s has a lowest ID than u’s, v becomes the cluster-head and adds a new entry to the dominators list, Dlist.
4. Upon receiving a cluster-head message, non-multicast node changes its state to VMB and broadcast its updated list to inform the VMB membership.
5. Upon receiving a cluster-head message multicast node becomes a cluster-head and inform its dominants by broadcasting its list.

In the second phase, we use the distributed Steiner approximation to compute the Steiner tree, T_{G_M}(V'_M), which connects the cluster-heads. Initially, the system is in waiting state. Algorithm starts by initializing given graph, source node, and table, TAB. Table, TAB, consists of four fields namely, tag, source, destination, and cost fields. Tag field is either ‘1’ or ‘0’ depending upon whether node is in the tree or not. Destination field depicts the tree node closest to the node in source field. The cost field represents the cost from the tree node to destination node. The pseudo-code of the approximation is as follows.

```
Initialize tree T_{G_M}(V'_M) and Table, TAB.
Select the source node, u, and T_{G_M}(V'_M) ← u
\(\backslash\) For the first cluster-head v ← T_{G_M}(V'_M) \(\backslash\)
Select cluster-head u_i ← T_{G_M}(V'_M)
\(\text{such that cost}(T_{G_M}(V'_M), v)\) is smallest
Change tag field: tag \(\backslash v\) ← ‘1’
Add to Routing table: ROUT_TAB ← v \cup ROUT_TAB
Update information in TAB
\(\backslash\) Start building a tree T_{G_M}(V'_M) \(\backslash\)
IF u \neq v in T_{G_M}(V'_M)
Pass the “message” to the next neighbor, v.
Check the status of v in TAB, tag field
IF tag-field[v]=0 \& cost field[v] is smallest
Add : T_{G_M}(V'_M) ← v \cup T_{G_M}(V'_M)
ELSEIF
∀ nodes u ∈ T_{G_M}(V'_M) All tag field [v]=1
⇒ all cluster-heads nodes are in T_{G_M}(V'_M) DONE!
```

Send completion message to all nodes in network
ELSE
Select the next neighbor, N[v], to go
Select u ∈ T_{G_M}(V'_M) \& (v ∈ T_{G_M}(V'_M)
\& tag field [v]=“0” \& cost field [v] = smallest
T_{G_M}(V'_M) ← v \cup T_{G_M}(V'_M)
Update ROUT_TAB

6 Conclusion
Ad hoc wireless network are highly dynamic networks, in which routing and clustering are important in determining the overall network performance. The virtual backbone approach is used in solving the routing problem in ad hoc wireless networks. Multicast routing is to find a tree rooted from a source node to all multicast destination nodes. In dominating set problem, we are required to find a minimum size subset of vertices with property that each vertex is either in the dominating set or adjacent to some vertex in the dominating set. In this paper, we concentrate on the related problem of finding a connected dominating set of minimum size in which the graph induced by vertices in the dominating set are required to be connected. Instead conventional spanning tree model, we used Steiner tree in unit-disk graph, which is NP-hard. The main contributions of this work are fully distributed Steiner tree with approximation ratio of 2 and fully distributed algorithm for multicast backbone structure with an approximation ratio at most 10. The main advantage of the algorithm is that it simplifies the routing process to one in a smaller subnetwork. V_M

References


