Multicasting under Multi-domain and Hierarchical Constraints

Prithwish Basu*, Chi-Kin Chau†‡, Richard Gibbens‡, Saikat Guha*, Ryan Irwin*
*Raytheon BBN Technologies, Cambridge, MA, USA
†Masdar Institute of Science and Technology, Abu Dhabi, UAE
‡University of Cambridge, Cambridge UK

***Abstract*—We consider the multicast routing problem under operational communication constraints, such as in practical deployment scenarios, e.g., multi-domain ad hoc networks where two or more teams form a coalition, or in tactical networks where information flows need to adhere to specified policies regardless of the physical connectivity of nodes. First, we consider the problem of minimum-cost multicast routing on a *multi-domain* network by constructing a node-weighted Steiner tree (for mesh networks) and a Steiner connected dominating set (for wireless broadcast networks) that is subject to a non-additive cost constraint. This is because the multi-domain multicast cost is not just the sum of node costs of a Steiner tree, but it instead depends on the domains of the connected neighbors. We give an efficient algorithm that provides an $O(\log k)$ approximation guarantee, where k is the number of terminal (or sink) nodes in the network. Taking multi-domain cost constraints into account can help reduce the cost of a multicast tree by up to 40%. We also consider a constraint imposed due to *hierarchy compliance*. We show that the overall multicast can be decomposed into several smaller multicasts, each of which might be efficiently solvable. We find that necessary hierarchical constraints could cause a significant increase in the total cost of multicast – up to 25%, as per our simulations based on a realistic deployment scenario.**

I. Introduction

Multicast networks are effective infrastructures for one-to-many and many-to-many communications, which can reduce the overhead of maintaining multiple one-to-one (i.e., unicast) communications, particularly in wireless networks. Typical multicast algorithms only provide basic connectivity to a given set of "terminals" in the network with minimum cost (on nodes and/or edges) while potentially using non-terminal nodes for relaying. In point-to-point connection networks such as mesh networks, the optimization problem is equivalent to finding a *minimum cost Steiner tree* (ST) that connects all the terminals [1]. On the other hand, in wireless broadcast networks, where a node could reach multiple neighbors simultaneously with a single broadcast, the problem is slightly different – it is equivalent to finding a *minimum cost Steiner connected dominating set* (StCDS) that connects all the terminals [2].

Corresponding author: pbasu@bbn.com. This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defense and was accomplished under Agreement Number W911NF-06-3-0002.l. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

Both ST [2], [3] and StCDS [2], [4] are well-studied in literature in the context of basic multicast. In this paper, however, we introduce two novel multicast network models where the constraints are imposed by (a) the *heterogeneous multi-domain* nature of the network, or (b) by *hierarchical policy-driven communications*.

In multi-domain multicast networks, the goal is to connect terminals belonging to multiple heterogeneous domains [5], which refer to logically or physically distinct groups of nodes. The presence of heterogeneous domains influences (and in fact constrains) connectivity among nodes. In particular, nodes can have heterogeneous transmission or forwarding costs that depend on the local connectivity to other nodes of different domains. Multicast forwarding in these networks typically involves higher-cost inter-domain gateway nodes, whose costs cannot be simply added to compute the total cost of the multicast subgraph (for both mesh and wireless broadcast network scenarios) – instead the total cost may depend on the particular configuration of domains at each node in the multicast subgraph. We aim to find optimal multicast subgraphs and associated domain configurations that minimize the transmission/forwarding costs across multiple domains.

A second type of constraint that affects multicast communication is that pertaining to information flow through hierarchies. For many operations and missions in practice, mere topological proximity to certain recipients of a message does not warrant direct delivery of the message to the latter. Instead, certain hierarchical policies that define different roles and ranks of network nodes may constrain the message flow through the network. For example, in military networks, communications between various nodes may need to be observed and then cleared by individuals located higher in the chain-of-command hierarchy. It is often the case that a subset of nodes in the hierarchy are interested in participating together in a multicast session. Therefore, we are motivated to construct multicast "routes" that *connect* these nodes while being constrained by the hierarchy of information flow.

The aforementioned constraints can significantly impact the routes chosen through multicast networks. In this paper, we study the underlying optimization problems that are applicable to constrained multicast in wireless networks both with and without wireless multicast advantage (WMA). We show that subtly different techniques can be applied to find guaranteed approximation algorithms for both the scenarios. Our contributions in this paper are as follows: (1) Optimization problem formulation of the multi-domain multicast problem with non-additive costs; (2) $O(\log n)$-approximation algorithms based on a node-weighted Steiner tree approximation algorithm and a Steiner CDS approximation algorithm for multi-domain costs; (3) Optimization problem formulation for multicasting under hierarchical constraints, and optimal and approximation algorithms for the same; (4) Extensive simulation based performance evaluation for both wireless mesh and broadcast networks. We show that taking multi-domain cost constraints into account can help reduce the cost of a multicast tree by up to 40%. We also show that necessary hierarchical constraints could cause a significant increase in the cost of multicast (up to 25% when measured in terms of total cost in our simulations that are based on a realistic deployment scenario).

II. Preliminaries and Related Work

Given a graph $G = (V, E)$, there is a set of *terminals* (or sinks) $M \subseteq V$ and $M \neq \varnothing$, which are required to participate in multicast communication. Let $|M| = k$. We typically denote $T = (V(T), E(T))$ as a subgraph of G, where $V(T) \subseteq V$ and $E(T) \subseteq E$.

We now describe below variants of the unconstrained multicast network optimization problem for both mesh and wireless broadcast networks.

A. Unconstrained Multicast Network Problems

Edge-weighted mesh network multicast Each edge $e \in E$ has an associated cost $c(e)$. $V \backslash M$ are called *Steiner nodes*, which may participate in multicast communication if their involvement can enable connectivity between terminals in M.

Definition 1: Edge-weighted multicast problem: $\min_T \sum_{e\in E(T)} c(e)$, subject to:

1) T is a connected subgraph of G, and
2) each pair of nodes $u, v \in M$ are connected via T.

The above problem (also known edge-weighted Steiner problem whose solution yields a tree T) has been studied extensively in the past. The problem is NP-hard not only for the general graph setting [6] but also for the special cases of Euclidean (L_2) norm [1] and Manhattan (L_1) norm [7]. It is well-known that many variants of edge-weighted multicast network (as known as Steiner network problems) have constant approximation algorithms [8], [9], based on LP relaxation and randomized rounding techniques.

Node-weighted mesh network multicast Each node $v \in V$ has an associated cost $c(v)$. Using previously defined notation, we define the following problem[1]:

Definition 2: Node-weighted multicast problem: $\min_T \sum_{v\in V(T)} c(v)$, subject to:

1) T is a connected subgraph of G, and
2) each pair of nodes $u, v \in M$ are connected via T.

[1]This is the node-weighted Steiner tree problem, which is a generalization of the edge-weighted Steiner problem since any instance of the latter can be converted to an instance of the former by a simple transformation that involves adding intermediate nodes to the graph for each edge.

Similarly, a Steiner tree problem for a graph with both node and edge weights (but additive) can be transformed to a purely node-weighted version.

For wireless networks with available Wireless Multicast Advantage (WMA), a node can reach all its 1-hop neighbors by a single broadcast under ideal channel conditions.

Node-weighted wireless broadcast This is equivalent to computing the minimum Connected dominating set (CDS) of G, which is a minimum node-cost connected subgraph such that *every node* is either in the subgraph or is a neighbor of some node in the subgraph [2], thus only those nodes in the CDS need to broadcast the message to their neighbors. Weights of "dominated" nodes not in the subgraph are not counted.

Node-weighted wireless multicast This is equivalent to computing the minimum Steiner connected dominating set (StCDS) of (G, M), which is a minimum node-cost connected subgraph such that *every terminal* is in the subgraph or is a neighbor of some node in the subgraph [2], thus only those nodes in the StCDS need to broadcast the message to their neighbors. Weights of "dominated" nodes (including terminals) in the subgraph are counted.

Figure 1 illustrates how ST and StCDS could yield significantly different results on the same input graph.

B. Approximation Algorithm for Node-weighted Steiner Trees

Klein and Ravi gave the first approximation algorithm for this problem with polynomial time complexity [3] – this yields a cost within a constant factor of the best-possible approximation algorithm, and has an approximation guarantee of a factor of $2(\log |M|)$ [3]. Note that it well known that the set-cover problem can be reduced to the node-weighted Steiner tree problem in an approximation-preserving manner, thus showing that the node-weighted Steiner tree problem is $\Omega(\log |M|)$-approximable.

To devise a tree construction algorithm for constrained multicast, we first discuss the Klein-Ravi algorithm for constructing a node-weighted Steiner tree [3], which forms a building block of our constrained multicast algorithms. The Klein-Ravi algorithm greedily and iteratively merges a set of subtrees to form a multicast tree. Initially, the set of subtrees consists of singleton sets of terminals $\mathcal{T}_{(0)} = \{\{v\} \mid v \in M\}$. At step t, the algorithm finds a node $v \in V$ and a subset of subtrees $F \subseteq \mathcal{T}_{(t-1)}$ such that the following distance metric is minimized: $\widetilde{\mathrm{dist}}(v, F) \triangleq \frac{c(v)+\sum_{T\in F} \mathrm{dist}(v,T)}{|F|}$. where $\mathrm{dist}(v, T)$ is the total node-weighted cost of a shortest path from v to reach the nodes in T, excluding the costs of the two end-nodes. Then, the subtrees in F will be merged with v using shortest paths to form one single subtree in $\mathcal{T}_{(t)}$. We summarize the Klein-Ravi algorithm below:

The running time of KleinRavi_Alg algorithm is $O(|V|^3 \log |V|)$, because finding the minimum $\widetilde{\mathrm{dist}}(v, F)$ among all F can be achieved by first sorting the set of shortest paths of every node to v in a decreasing order, and then inserting each subtree incrementally into F to determine the minimum $\widetilde{\mathrm{dist}}(v, F)$ (see [3]).

Algorithm 1 KleinRavi_Alg$[G, M, c(\cdot)]$

Set $\mathcal{T}_{(t)} = \{\{v\} \mid v \in M\}$
$t = 0$
while $|\mathcal{T}_{(0)}| > 1$ **do**
  $t \leftarrow t + 1$
  Find $v \in V$ and $F \subseteq \mathcal{T}_{(t-1)}$ such that $\widetilde{\text{dist}}(v, F)$ is minimized
  $T' \leftarrow$ subtree by merging F with v using shortest paths from v
  $\mathcal{T}_{(t)} \leftarrow (\mathcal{T}_{(t-1)} \backslash F) \cup \{T'\}$
Output $\mathcal{T}_{(t)}$

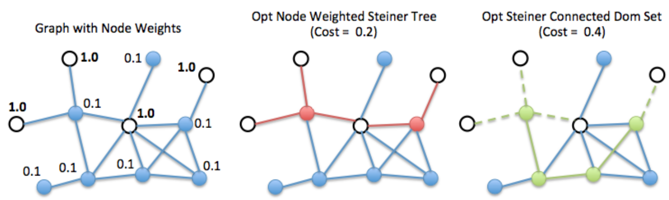

Fig. 1. Node-weighted Steiner Trees and Steiner connected dominating sets ("terminal" nodes are denoted by unfilled circles)

Since it is possible to transform a graph with both node and edge costs to one with only node costs, we only consider the purely node-weighted version of the problem in this paper[2].

III. Problem Formulation

In the following subsections, we present the mathematical formulations for two constrained multicast network problems.

A. Multi-Domain Multicast

We first consider a set of domains $\mathcal{D}$. We define $\forall_{v\in V} \mathcal{D}_v \subseteq \mathcal{D}$ as the set of possible domains for v. Each node may configure its associated domains as any subset of $\mathcal{D}_v$.

We next define a network-wide *feasible configuration of domains* that is subject to the constraint that a pair of nodes can directly communicate with one another if the nodes are configured with certain common domains. Given a subgraph T, we define a feasible configuration of domains w.r.t. T as $(x_T(v) \subseteq \mathcal{D}_v)_{v\in T}$, such that if $(u, v) \in E(T)$ then $x_T(u) \cap x_T(v) \neq \varnothing$. Given a feasible configuration of domains x_T, each node $v \in V$ has an associated cost $c(v, x_T(v))$, which is a function of the configured domains $x_T(v)$.

Definition 3: Multi-domain multicast problem: $\min_{x_T, T} \sum_{v\in V(T)} c(v, x_T(v))$ subject to:

1) T is a connected subgraph of G, and
2) each pair of nodes $u, v \in M$ are connected via T, and
3) x_T is a feasible configuration of domains w.r.t. T.

We note that the set of Steiner (or non-terminal) nodes can be regarded as "gateways" to bridge the connectivity across multiple domains using appropriate domain configurations. We have to optimize the selection of Steiner nodes as well as their domain configurations for constructing a multicast network.

Further, we assume the following *monotonicity* and *submodularity*[3] constraints on $c(v, \cdot)$ for all $v \in V$ and any $D \subseteq \mathcal{D}_v$:

$$\max_{d\in D} c(v, d) \leq c(v, D) \leq \sum_{d\in D} c(v, d) \tag{1}$$

[2]While there are more recent algorithms that improve upon $O(\log |M|)$ approximation by a constant factor, e.g., [2], we decided to use the Klein-Ravi algorithm for our investigations, as the latter is simpler.

[3]Note that this is similar to, but not the same as the full definition of submodularity in combinatorics.

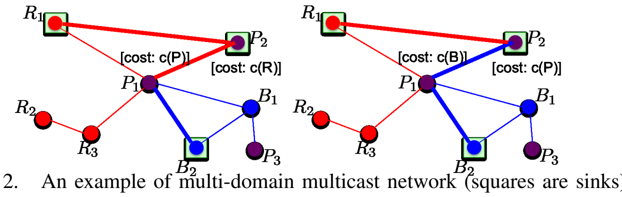

Fig. 2. An example of multi-domain multicast network (squares are sinks)

This is a natural constraint because $c(v, \cdot)$ should be greater than the maximum cost of individual domains, and should not exceed the total cost of individual domains. Indeed, in several realistic application scenarios, the marginal cost associated with a larger set diminishes with set size.

Figure 2 illustrates the above with a simple example. Consider a multi-channel network (e.g., radio or optical), where each node is equipped with transceivers tuned to one of more frequencies. These frequencies can be considered as domains. Suppose $\mathcal{D} = \{R, B\}$, where R and B indicate red and blue domains. The possible domains for the one-frequency nodes R_i's and B_i's are $\mathcal{D}_v = \{R\}$ and $\mathcal{D}_v = \{B\}$, respectively, whereas the possible set of domains for the two-frequency nodes P_i's is given by $\mathcal{D}_v \in \{\{R\}, \{B\}, \{R, B\}\}$. Analogously, we refer to the domain $\{R, B\}$ as the "purple" (P) domain in this example. The cost of the purple domain $c(P) = c(\{R, B\})$ characterizes the cost to process data encoded in input red (or blue) frequency and output data in both red and blue frequencies. Also, $c(R)$ denotes the cost to process "red" data alone (and similarly for $c(B)$).

Figure 2 shows two "domain configurations" at purple nodes P_1, P_2 and a Steiner tree which connects the sink nodes (marked in squares). In the first configuration, $\mathcal{D}_{P_1} = \{\{R, B\}\}$, whereas $\mathcal{D}_{P_2} = \{\{R\}\}$. Therefore, $c(P_1) = c(P)$ and $c(P_2) = c(R)$. Similarly, in the second configuration, $\mathcal{D}_{P_1} = \{\{B\}\}$, whereas $\mathcal{D}_{P_2} = \{\{R, B\}\}$. Therefore the costs of the two topologically isomorphic Steiner trees are different, i.e., $2c(R)+c(B)+c(P)$ and $c(R)+2c(B)+c(P)$, respectively. In contrast, a standard node-weighted Steiner tree would have a higher cost, $c(R) + c(B) + 2c(P)$. Our goal is to find the lowest cost tree and the corresponding configuration efficiently (since in general, there are an exponentially large number of configurations for a tree).

Multi-domain wireless multicast: The wireless network scenario has a natural analog to the multi-domain multicast problem. Instead of finding a Steiner tree, the problem here is to find an appropriate Steiner connected dominating set with a suitable domain configuration that minimizes the total cost of the StCDS.

Let $\Delta_T(v)$ be the degree of node v in a subgraph T.

Definition 4: Multi-domain Steiner CDS: $\min_{x_T, T} \sum_{v\in V(T):\Delta_T(v)>1} c(v, x_T(v))$ subject to:

1) T is a connected subgraph of G, and
2) each pair of nodes $u, v \in M$ are connected via T, and
3) x_T is a feasible configuration of domains w.r.t. T.

B. Hierarchy-Compliant Multicast

For the second problem, we consider a hierarchy of roles of network nodes that can be captured by an acyclic graph $H = (V_H, E_H)$, and a mapping $h(\cdot) : V \mapsto V_H$.

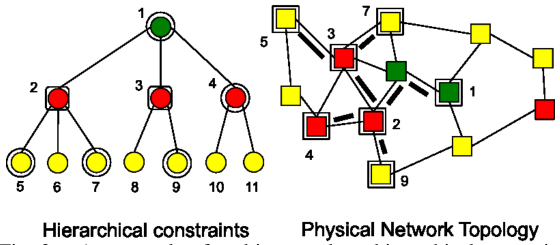

Fig. 3. An example of multicast under a hierarchical constraint.

A path in graph can be regarded as a sequence of nodes in the order it traverses. Given a path P in G, we write $h(P)$ as the sequence of the corresponding roles in H. A sequence Q is said to be a subsequence of Q', if removing some elements in Q' can create Q.

A subgraph T is said to be H*-compliant*, if any pair of nodes in M are connected in T, then there exist a path P in T and a path Q in H, such that Q is a subsequence of $h(P)$.

Definition 5: Hierarchy-compliant multicast problem: $\min_T \sum_{v\in V(T)} c(v)$, subject to

1) T is an H-compliant connected subgraph of G, and
2) each pair of nodes $u, v \in M$ are connected via T.

Figure 3 illustrates this class of constrained multicast networks with an example. An organizational hierarchy is shown on the left, and the corresponding physical deployment topology is shown on the right. Node 1 (root of the hierarchy) wants to multicast data to a set of nodes $\{4, 5, 7, 9\}$ (circles). However, the hierarchical constraints constrain the flow of the data as the "supervisors" of these aforementioned nodes must act as intermediaries (nodes $\{2, 3\}$ here). This may be necessary as the latter have better context about the subordinates than the root of the organization and they may want to even modify or embellish the message. Therefore, not only does the multicast sink set have to expand to include these intermediaries, the order in which the message is delivered to the sinks must obey the hierarchy. This may cause the information flow to get "stretched", and hence we need efficient algorithms for finding low cost multicast structures that strike the balance between hierarchical constraints and exploit the properties of the physical network.

Hierarchy compliant wireless multicast: This is a natural analog for the wireless case where WMA can be leveraged.

Definition 6: H-compliant wireless multicast problem: $\min_T \sum_{v\in V(T)} c(v)$, subject to

1) T is an H-compliant connected subgraph of G, and
2) $u \in M$ is either in T or is a neighbor of a node in T.

Although our formulation captures general hierarchies, most practical policy-driven hierarchies are tree based (e.g., command-and-control hierarchy). Hence, in this paper, we will focus on the construction of hierarchy-compliant multicast networks with respect to tree hierarchy.

IV. Multi-Domain Multicast Algorithms

In this section, we present approximation algorithms for the multi-domain multicast problem. We first consider the mesh network case and then the wireless broadcast network case.

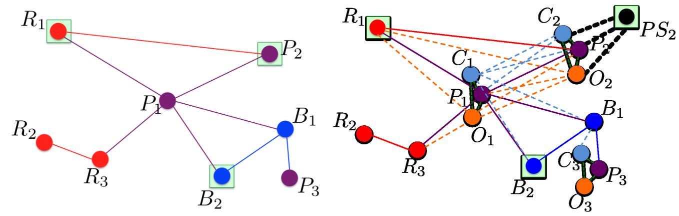

Fig. 4. Graph transformation to support non-additive multicast costs: (Left) Original graph, G. (Right) Augmented graph, G'. Multicast terminal nodes are denoted by green squares. New black nodes (PS_2) correspond to multi-domain nodes that are terminals. New edges (shown as dashed or in dark color) are drawn according to the graph augmentation rules. Trivial augmentations of single-domain terminals (R_1, B_1) are not shown for the sake of reducing clutter.

A. Mesh Networks: Multi-domain Steiner Tree

We rely on a gadget called *augmented graph*. Given a graph $G = (V, E)$ and $(\mathcal{D}_v)_{v\in V}$, we define an augmented graph $G_{\text{aug}}(M) = (V_{\text{aug}}, E_{\text{aug}})$ as follows:

1) $V_{\text{aug}} \triangleq \{v_D \mid v \in V, D \in \mathcal{D}_v\} \cup M$
2) $E_{\text{aug}} \triangleq E_{\text{dom}} \bigcup E_{\text{prxy}} \bigcup E_{\text{ter}}$, where

$$\begin{aligned} E_{\text{dom}} &= \big\{(u_D, v_{D'}) \mid (u, v) \in E, \\ &\qquad D \in \mathcal{D}_u, D' \in \mathcal{D}_v, D \cap D' \neq \varnothing\big\} \\ E_{\text{prxy}} &= \big\{(v_D, v_{D'}) \mid v \in V, D \in \mathcal{D}_v, D' \in \mathcal{D}_v\big\} \\ E_{\text{ter}} &= \big\{(v, v_D) \mid v \in M, D \in \mathcal{D}_v\big\} \end{aligned}$$

In the above steps, we replicate the connectivity in the original graph G and augment the connectivity considering every possibility of domain configurations between a pair of neighboring nodes. Also, we duplicate every terminal in $G_{\text{aug}}(M)$ as a unaltered terminal and a set of proxy nodes with all possibilities of domain configurations. We present an example of augmented graph generation in Figure 4.

We present our approximation algorithm in Algorithm 2. Our algorithm first applies KleinRavi_Alg to the augmented graph G_{aug}. Then, we project the output from KleinRavi_Alg to a valid Steiner tree in the original graph G.

Algorithm 2 MDMN_Alg$[G, M, \big(\mathcal{D}_v, c(v,\cdot)\big)_{v\in V}]$

```
Construct G_aug(M) from G, M and (D_v)_{v∈V}
for all v_D ∈ V_aug \ M do
    Set c'(v_D) = c(v, D)
for all v ∈ M do
    Set c'(v) = 0
T' ← KleinRavi_Alg[G_aug, M, c'(·)]        ▷ Apply Klein-Ravi algorithm
                              ▷ Now, map to a feasible configuration in G
Set T = (V(T) = ∅, E(T) = ∅)
for all v_D ∈ V(T') do
    V(T) ← V(T) ∪ {v}
for all (u_D, v_D') ∈ E(T') do
    E(T) ← E(T) ∪ {(u, v)}
for all v ∈ V(T) do
    Set x_T(v) = ⋃_{v_D ∈ V(T')} D
Output (T, x_T)
```

Since the augmentation steps are meant to produce a graph $G_{\text{aug}}(M)$ that duplicates the connectivity of neighboring nodes for every feasible configuration of domain, the optimal Steiner tree on $G_{\text{aug}}(M)$ naturally corresponds to the optimal Steiner tree on multi-domain G. However, this is not trivial, because the mapping between a Steiner tree on $G_{\mathsf{aug}}(M)$ and the one on G is not always bijective.

Theorem 1: An optimal node-weighted Steiner tree on the augmented graph $G_{\mathsf{aug}}(M)$ (i.e. the output of running the exact Steiner tree construction on it) has the same cost as an optimal Steiner tree on G considering multi-domain constraints.

Proof: We first define a mapping $f(T') \mapsto (T, x_T)$, where T' is a Steiner tree on $G_{\mathsf{aug}}(M)$ and T is a Steiner tree on G with configuration of domains x_T, such that

- If $v_D \in V(T')$, then we have $v \in V(T)$.
- If $(v_D, u_{D'}) \in E(T')$, then we have $(u, v) \in E(T)$.
- Set $x_T(v) = \bigcup_{v_D \in V(T')} D$

Note that the definition of mapping $f(\cdot)$ is the same as the corresponding mapping in MDMN_Alg.

Step 1: We first show that $f(\cdot)$ produces a feasible configuration x_T w.r.t. T. Otherwise, there exists a violated edge (u, v) such that $x_T(v) \cap x_T(u) = \varnothing$. Note that the construction of $G_{\mathsf{aug}}(M)$ requires that any $(u_D, v_{D'}) \in G_{\mathsf{aug}}(M)$ satisfies $D \cap D' \neq \varnothing$. Hence, (u, v) will not be an edge in T. This generates a contradiction.

Step 2: Suppose T' is an optimal Steiner tree on $G_{\mathsf{aug}}(M)$. We then show that the total node-weighted cost of T' is equivalent to that of T. Note that by $f(\cdot)$, $v_D \in V(T')$ iff $v \in V(T)$. We prove by contradiction. Suppose that the total node-weighted cost of T is not equivalent to that of T'. Then there exists a collection $\{D_i \subseteq \mathcal{D}_v\}$ such that $v_{D_i} \in V(T')$ and $\bigcup_i D_i = x_T(v)$ and $c(v, x_T(v)) \neq \sum_i c(v, D_i)$. Also, by submodularity constraint (Eqn. (1)), we obtain

$$c(v, x_T(v)) \leq \sum_{d \in x_T(v)} c(v, d) \leq \sum_i c(v, D_i) \qquad (2)$$

Hence, $c(v, x_T(v)) < \sum_i c(v, D_i)$. Since $v_{x_T(v)}$ has the same connectivity as the the union of $\{v_{D_i}\}$. Replacing $\{v_{D_i}\}$ by a single node $v_{x_T(v)}$ in T' will not change the connectivity, however can give a strictly lower cost. This generates a contradiction that T' is an optimal Steiner tree on $G_{\mathsf{aug}}(M)$.

Step 3: Suppose T' is an optimal Steiner tree on $G_{\mathsf{aug}}(M)$. We show that T is an optimal Steiner tree on G considering multi-domain constraint. Suppose T is not optimal Steiner tree, and there exists $\hat{T}$ on G, which has a lower node-weighted cost. We create an augmented graph $\hat{T}_{\mathsf{aug}}$ by $\hat{T}$. Note that $\hat{T}_{\mathsf{aug}}$ is a subgraph of $G_{\mathsf{aug}}(M)$. The cost is preserved in augmented graph. Hence, the node-weighted cost of $\hat{T}_{\mathsf{aug}}$ is the same as that of $\hat{T}$, but is lower than that of T'. This generates a contradiction that T' is an optimal Steiner tree on $G_{\mathsf{aug}}(M)$.

Finally, we note that every terminal has a corresponding unaltered terminal and a set of proxy nodes with all possibilities of domain configurations in $G_{\mathsf{aug}}(M)$. The unaltered terminal carries zero cost. Hence, this will not increase the cost of optimal Steiner tree on $G_{\mathsf{aug}}(M)$.

Therefore, by **Steps 1-3** we conclude that an optimal node-weighted Steiner tree on the augmented graph $G_{\mathsf{aug}}(M)$ has the same cost as an optimal Steiner tree on G. ■

Theorem 2: Algorithm MDMN_Alg produces an $O(\log|M|)$-approximation to an optimal Steiner tree on G considering multi-domain constraint.

Proof: By Theorem 1, if we replace KleinRavi_Alg in MDMN_Alg by exact construction of Steiner tree, then we can obtain an optimal Steiner tree on G considering multi-domain constraint. Since KleinRavi_Alg is an $O(\log|M|)$-approximation to an Optimal_Steiner on a unconstrained graph, and by Theorem 1 **Step 1**, we show that MDMN_Alg produces a feasible configuration x_T w.r.t T. Hence, Algorithm MDMN_Alg produces an $O(\log|M|)$-approximation to an Optimal_Steiner on G compliant with the multi-domain constraint. ■

Note that since we assume $|\mathcal{D}|$ is a constant independent of $|V|$, MDMN_Alg still runs in polynomial time.

B. Wireless Broadcast Networks: Multi-domain Steiner CDS

We now present an algorithm for approximating multi-domain CDS. The graph augmentation technique of Sec. IV-A is unsuitable since only the cost of leaf terminals is not counted while finding an StCDS, thus prompting another approach. First, greedily pick nodes that cover more than one of the previously uncovered terminals. For those covered elements, pick a representative terminal out of each greedy cover together with uncovered terminals (B). Next construct a (uniform) min-edge Steiner tree over B. Finally, greedily configure domains over these nodes in the Steiner tree.

First, define set cover problem as a set $\mathcal{U}$ of N items, and a family of subsets (called covers) $\mathcal{S} \subseteq 2^{\mathcal{U}}$. We aim to pick the a subset of covers $\tilde{\mathcal{S}} \subseteq \mathcal{S}$ such that $\min_{\tilde{\mathcal{S}} \subseteq \mathcal{S}} |\tilde{\mathcal{S}}|$, subject to the constraint of covering all items $\bigcup_{C \in \tilde{\mathcal{S}}} S = \mathcal{U}$. We present a modified greedy set cover algorithm in Algorithm 3, which stops when no more than one uncovered item can be covered.

Algorithm 3 GreedySetCover$[\mathcal{U}, \mathcal{S}, c(\cdot)]$

$\tilde{\mathcal{S}} \leftarrow \varnothing$, $C \leftarrow \varnothing$, $A \leftarrow \varnothing$, $\mathsf{S} \leftarrow \mathcal{U}$
repeat
  Choose $\mathsf{S} \in \mathcal{S}$ with minimal $\sum_{i \in \mathsf{S} \setminus C} c(i)$ ▷ break ties arbitrarily
  if $|\mathsf{S} \setminus C| > 1$ **then**
    $A \leftarrow A \cup \{u\}$ for some $u \in \mathsf{S} \setminus C$
    $C \leftarrow C \cup S, \tilde{\mathcal{S}} \leftarrow \tilde{\mathcal{S}} \cup \{\mathsf{S}\}$ ▷ Update set of covered elements
until $|\mathsf{S} \setminus C| \leq 1$ or $C = \mathcal{U}$
return $(\tilde{\mathcal{S}}, A)$

Define $\mathcal{N}(v) \subset V$ as the set of neighbors of $v \in V$.

Algorithm 4 MStCDS_Alg$[G, M, c(\cdot)]$

for $v \in V$ **do**
  Create $S_v \leftarrow M \cap \mathcal{N}(v)$
$(\tilde{\mathcal{S}}, A) \leftarrow$ GreedySetCover$[M, \{S_v \mid v \in V\}, c(\cdot)]$
$U \leftarrow M \backslash (\bigcup \tilde{\mathcal{S}})$
Construct edge-minimizing Steiner tree T over terminal set $U \cup A$
$B \leftarrow \{v \mid S_v \in \tilde{\mathcal{S}}\}$
Set StCDS as $T \leftarrow \mathsf{T} \cup B$
for $v \in T$ traversing from leaf nodes to upstream **do**
  if v is a leaf node **then**
    Set $x_T(v) \leftarrow \mathcal{D}_v \cap \mathcal{D}_u$, where $u \leftarrow \mathrm{parent\_of}(v)$
  else
    Set $x_T(v) \leftarrow \cup_{w \text{ is\_child\_of } v}\, x_T(w)$
Output (T, x_T)

Theorem 3: MStCDS_Alg (Algorithm 4) yields an approximation factor $O(\frac{\max_{v \in V, D \subseteq \mathcal{D}_v} c(v,D)}{\min_{v \in V, d \in \mathcal{D}_v} c(v,d)} \log(|V|))$.

Proof: We extend the proof of Guha-Khuller [2] to the multi-domain setting. First, we note that $|B| = |Opt| \log |V|$, where Opt is the optimal set of CDS without considering multi-domain constraint. Also, note that $|U| \leq |Opt|$ because no more than one node in U can be covered by one node in CDS (otherwise, such nodes cannot be in U). Then there exists a Steiner tree connecting $U \cup A$ using at least $|U|+|A|+|Opt|-1$ edges (i.e. $|U|+|A|$ edges for connecting to $U \cup A$, while $|Opt|-1$ edges connecting to the nodes in tree of size $|Opt|$). We note that there exists a constant approximation algorithm for edge-minimizing Steiner tree (See [2]). Hence, we obtain an approximate edge-minimizing Steiner tree of at most size $c \cdot (|U| + |A| + |Opt| - 1)$. This gives a CDS with at most $c \cdot (|U| + |A| + |Opt| - 1) + |B|$. Therefore, the size of CDS is upper bounded by $O(\log |V| \cdot |Opt|)$.

Finally, we configure the domains to satisfy the multi-domain constraint by $x_T(v) \leftarrow \cup_{w \text{ is\_child\_of } v} x_T(w)$. The optimal CDS considering multi-domain constraint will have at least $|Opt|$ nodes. The worst case of multi-domain configuration will incur at most a cost of a factor $\frac{\max_{v \in V, D \subseteq \mathcal{D}_v} c(v,D)}{\min_{v \in V, d \in \mathcal{D}_v}}$. Hence, this completes the proof. ■

V. Hierarchy-compliant Multicast Algorithms

In this section, we give efficient algorithms for multicasting under constraints imposed by a logical hierarchy in which the network users are embedded. In this model, a node can *read* the message content only if it is signed by its parent node in the hierarchy. Nodes can, however, freely route messages in the network without *reading* their contents.

We consider the special case where the acyclic graph $H = (V_H, E_H)$ mentioned in Section III-B is in fact a tree and the function $h(\cdot)$ is bijective. An edge $(u,v) \in E_H$ means that node v can receive a message from node u only after it is *signed* (or authenticated) by u. Without such authentication, the message can be neither *read* nor *modified* by v, although it can be *forwarded* along.

As mentioned in Section III-B, the Hierarchy Compliant Multicast (HCM) problem involves augmenting the set of terminal nodes M by their ancestors in H. In particular, $M' = M \cup \{u \mid \forall s \in M, u = ancestor(s,H)\}$. We consider two models here.

We first consider the caching-friendly version of the problem (CFM) which is more straightforward. If caching the message is allowed and the message is not modifiable, it may be distributed via Node-weighted Steiner multicast and cached before the *authentication/authorization* occurs. The latter can happen in hierarchical precedence order starting from the root node R towards the respective leaves in M along the edges of the tree H. The control overhead of the authentication process is typically negligible compared to that of the overhead of distribution of the actual data.

In the *strict-precedence* version of the problem, ancestor nodes cannot receive the message *before* their subordinates receive it. This phenomena could indeed occur if we flat out ran KleinRavi_Alg (Algorithm 1) on (G, M'). We also observe that in order to adhere to the strict precedence constraints, a message may have to traverse certain vertices and edges more than once. For example, if $(u - v - w)$ is a fragment of communication graph G and the message m currently resides at u and v, and the precedence constraints demand that $\{(h(v), h(u)), (h(u), h(w))\} \subseteq E_H$, then after receiving m from v, u will transmit a copy of m toward w, thus traversing the edge (u,v) again. This may be unavoidable if u desires to make modifications to m before forwarding it to w.

HCM cannot be directly mapped to a Steiner tree problem. In fact the lowest cost topological structure need not necessarily be a tree. The steps to compute optimal and approximate HCM structures are listed in Algorithm 5.

Algorithm 5 BFSteiner_Alg$[G, M, H, R \in V, c(\cdot)]$

1: $M' = M \cup \{u \mid \forall s \in M, u = ancestor(s,H)\}$
2: $H' = induced\_subtree(M', H)$
3: $C \leftarrow \{c \mid c \in children(R, H')\}$ ▷ breadth-first traverse H' from R
4: $S_R \leftarrow$ Steiner_Alg$[G, \{R\} \cup C, c(\cdot)]$ ▷ Solve mini-Steiner tree problem. Steiner_Alg could be Optimal_Steiner or KleinRavi_Alg
5: **while** $C \neq \varnothing$ **do** ▷ BFS and solve mini-Steiner tree problems
6: $\quad c \leftarrow popfront(C)$
7: $\quad S_c \leftarrow$ Steiner_Alg$[G, \{c\} \cup children(c, H'), c(\cdot)]$
8: $\quad pushback(C, children(c, H'))$
9: $SubG \leftarrow \bigcup_{u \in H'} S_u$ ▷ Union of Steiner trees

Note that it is possible for solve HCM optimally in a reasonable amount of time (unlike CFM or even MDMN) when the degree of H is constant and does not grow with $|V|$ or $|M|$. This is because the overall optimization problem can be decomposed into multiple *smaller sized* Steiner tree problems, each of which can be solved optimally (and independently) using an Integer Programming formulation (Optimal_Steiner). At lines 4 and 7 in Algorithm 5, although the number of nodes in the graph is $|V|$, the number of terminals is typically much smaller than $|M'|$ and is bounded by the out-degree of H', which is in turn bounded by the out-degree of H.

Theorem 4: BFSteiner_Alg$[G, M, H, R, c(\cdot)]$ returns the optimal subgraph T of G when Optimal_Steiner is used.

Proof: We use notation defined in Section III-B. H' is the tree rooted at R induced on H by the multicast terminal set M'. Consider an arbitrary subtree T' of H' which is rooted at node $r \in V_H$. Let $C_{H'}(r)$ be the direct children of r in H'. Any optimal subgraph $SubG$ that connected all terminals in M in an H-compliant manner must contain a subgraph S_u of G that connects node $u \in V_G$ (such that $h(u) = r$) with certain nodes $v_1, v_2, \ldots, v_{|C_{H'}(r)|} \in V_G$ such that $\{h(v_1), h(v_2), \ldots, h(v_{|C_{H'}(r)|})\} = C_{H'}(r)$. Now, S_u can be computed independently of other such subgraphs that are computed during the execution of BFSteiner_Alg because none of the portions of the previously computed subgraphs can be reused during the computation of S, for if these portions are reused, there will be a violation of H-compliance. Moreover, there will be violation of H-compliance if one or more of nodes in set $\{u, v_1, v_2, \ldots, v_{|C_{H'}(r)|}\}$ do not get included in T. This follows from the fact that $h(\cdot)$ is bijective. Since S_u can be computed independently, if an optimal Steiner tree algorithm is used to compute it, then $SubG = \bigcup_{u \in M'} S_u$. ■

Corollary 5: If H has maximum degree k,

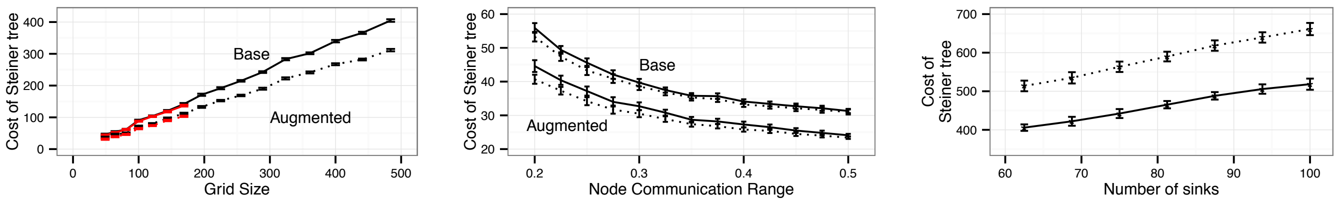

Fig. 5. Scaling of multicast tree costs: (a) Grid networks – optimal curves plotted for $N \leq 196$; (b) RGG – optimal (dotted) and Klein-Ravi approximation (solid); (c) vs. number of terminals (for $N = 625$) – base graph (dotted); augmented graph (solid)

$\mathsf{BFSteiner\_Alg}[\mathsf{G}, \mathsf{M}, \mathsf{H}, \mathsf{R}, \mathsf{c}(\cdot)]$ returns a subgraph T with cost within $2\log(k+1)$ of that of the optimal subgraph T, when $\mathsf{KleinRavi\_Alg}$ is used.

Proof: Since the maximum number of terminals in each "mini" Steiner tree computation is $k+1$, according to Klein-Ravi's approximation guarantee of $2\log(k+1)$, the cost of each mini-Steiner tree is within at most $2\log(k+1)$ factor of the optimal. Therefore the total cost obeys the same bound. ∎

H-compliant wireless multicast $\mathsf{BFSteiner\_Alg}$ can be naturally extended to the wireless network setting. Essentially, instead of repeatedly computing Steiner trees that connect H-compliant nodes at consecutive levels in H, one needs to repeatedly compute Steiner CDSes that can connect the H-compliant nodes at consecutive levels in H. In particular, Lines 4 and 7 in Algorithm 5 need to be replaced by a call to either the Optimal Steiner CDS algorithm or to Guha and Khuller's approximation algorithm for computing Steiner CDS [2].

HCM is significantly different from overlay multicast [10]. In the latter, the overlay network is used to perform point-to-point unicast to achieve multicast, mainly because multicast is not widely supported in the underlay network. In contrast, in our setting, multicast is available as a primitive and the precedence constraints impose an overlay structure on top of the multicast-capable network substrate. Hence the solution space for our algorithms is likely to be richer.

VI. Performance Evaluation and Discussion

Multi-domain multicast We first evaluate the performance of the multi-domain multicast algorithm by extensive simulations on both grids and random geometric graphs (RGG). Each node is assumed to belong to R or B domains with probabilities p_r and p_b, respectively, thus a node is a (purple) gateway node with probability $p_p = 1 - p_r - p_b$. Then, edges are created between adjacent nodes, with the only forbidden pair being *red-blue*. Finally, nodes act as multicast terminals with probability f. For each instance of a randomly generated grid graph (base G), we apply the augmentation rules given in Section IV to generate the augmented graph G_{aug}, and then run $\mathsf{Optimal\_Steiner}$ as well as Algorithm 1 on both G and G_{aug}. The objective is to characterize the gain achieved by exploiting the fact that a purple node may be charged only c_b (c_r) if it is only communicating with blue (red) nodes.

Initially, we vary the network size $N = n^2$ for fixed parameters ($p = 1.0, p_r = 0.2, p_b = 0.3, f = 0.1, c_r = 2, c_b = 3, c_p = 4$). We first observe in Fig. 5(a) (all error bars indicate 95% confidence intervals) that it is feasible to run the Optimal algorithm only up to $k \approx 14$ terminals; beyond that, we experience exponentially large running times. However, we observe that $\mathsf{KleinRavi\_Alg}$ is able to return very good approximations of the Optimal, much lower than the worse case approximation factor of $2\ln k$ for both base and augmented graphs. We also observe that the average multicast tree cost increases linearly with N – this is because the terminals are uniformly randomly distributed across the network (therefore, the number of nodes in ST scales linearly with N), and node costs c_r, c_b, c_p are not very different.

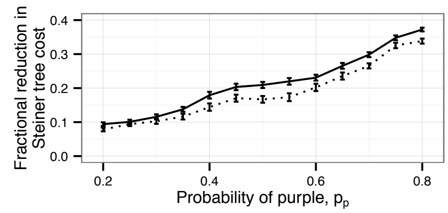

Fig. 6. Fractional difference in multicast tree costs on 2D grids for Optimal (solid curve) and Klein-Ravi (dotted curve).

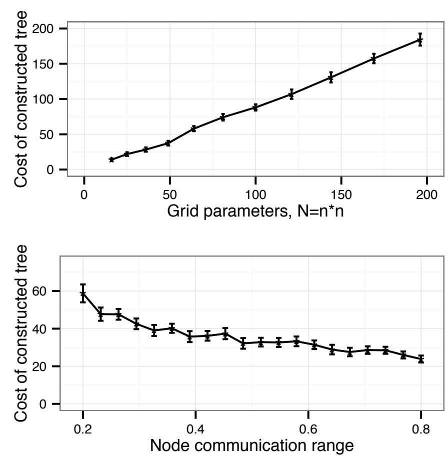

Fig. 7. Multicasting with WMA: (a) wireless grid network; (b) 100 node RGG. Cost is that of StCDS + leaf nodes.

A significant finding is that the Steiner tree costs on G are higher than those on G_{aug} because in the latter, it is possible to find paths that pass through purple gateway nodes (P) but with no $R \leftrightarrow B$ conversions. Interestingly, the fractional decrease in costs hovers between $20 - 30\%$ over the entire range of N for the case of Optimal and stabilizes around 25% for the case of the Klein-Ravi. Figure 5(b) illustrates the relative performance on RGGs. We observe that as the connectivity range is increased, the Steiner tree cost decreases since a fewer number of non-terminal nodes are needed to connect the terminals. However, we observe that the gains

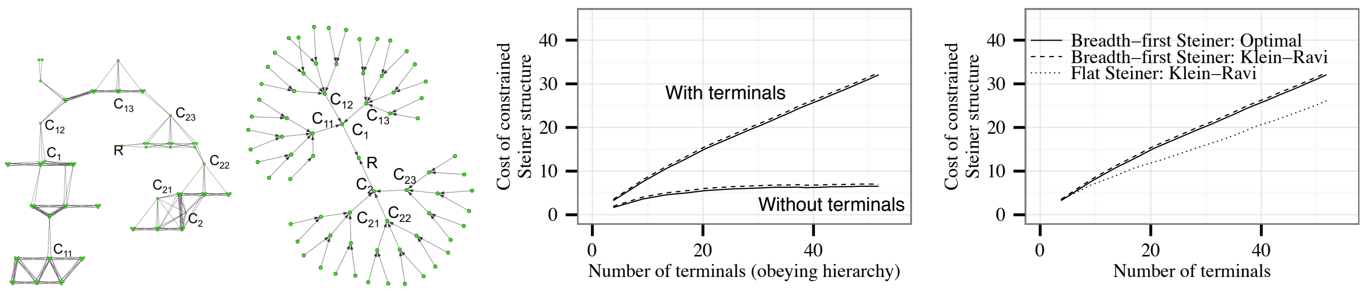

Fig. 8. H-compliant multicast: (a) Historical deployment and hierarchy; (b) Costs: Klein-Ravi (dotted), Opt (solid); (c) With and without H-compliance

due to augmentation hold steady for both Optimal and Klein-Ravi. If f is varied while keeping N fixed, we observe a roughly linear scaling trend (In Figure 5(c)). The terminals are distributed uniformly randomly which results in the number of nodes in the Steiner tree growing almost linearly. We also obtain over 25% cost savings due to augmentation.

In Figure 6, we examine the impact of varying the fraction of multi-domain nodes on the fractional difference between the multicast tree costs on base and augmented graphs. As p_p is increased, for $N = 100, f = 0.1$, we observe that the fractional difference increases significantly (up to 40%). This confirms our hypothesis that classic multicast (which sums the cost of multi-domain nodes, irrespective of whether they are performing inter-domain message conversions) is unable to exploit the opportunities to find low cost multicast trees; and thus the proposed graph augmentation is necessary.

Figure 7 shows the performance of Algorithm 4 when WMA is available. We observe a similar trends of cost vs. network size or Tx range as in the case with no WMA. The StCDS costs in Fig. 7 are higher than corresponding ST costs in Fig. 5 (both include terminal costs). A likely explanation is that while ST minimizes the total number of nodes in the tree connecting the terminals, StCDS minimizes only the number of non-leaf nodes in the CDS connecting the terminals.

Hierarchy-compliant multicast We use a 91 node synthetic data set based on a historical military deployment scenario where nodes in the network were tagged with geolocation and hierarchy labels[4]. Figure 8(a) illustrates the locations of various troop units and their relative positions in the military hierarchy. We simulated random sets of multicast terminals M in this network and calculated the cost of H-compliant multicast. The node costs were also chosen randomly $\in (0, 1]$. Obviously, the requirement of H-compliance results in a significant increase in the number of terminals (the new ones are ancestors of M in H). We observe from Figure 8(b) that as $|M|$ grows, the cost of the Steiner subgraph begins to get dominated by the costs of the terminal nodes themselves. This is because, if there are many terminals, most terminals do not need help from non-terminals to get connected to each other. Also, the costs yielded by both Optimal and Klein-Ravi versions of Algorithm 5 are very close to each other, and since Optimal does not take longer than Klein-Ravi to run when executed at each level of H (and thus with a bounded number of terminals), either algorithm is a reasonable candidate to be used within Algorithm 5.

Finally, we illustrate the overhead of H-compliance on multicast in Figure 8(c). We executed Algorithm 5 (with both Optimal and Klein-Ravi variants) as well as flat Klein-Ravi Steiner tree algorithm without H-compliance (for the CFM algorithm mentioned in Section V) on the 91 node network. The latter obviously yields significantly lower costs since the flows do not have to be H-compliant. However, if H-compliance is a requirement, then one must be willing to pay a 25% overhead in total costs. If only non-terminal costs are measured, then the relative overhead is much more significant.

Although hierarchical constraints on multicast can result in significant *stretch* of information flows thus resulting in significant increase in costs of Steiner structures connecting the terminals, the good news is that approximation algorithms (with worst case guarantees) can perform close to their optimal counterparts, while executing in low polynomial time.

[4]This data set was generated from sources in the public domain: British Army website: http://www.army.mod.uk; Ministry Of Defense website: http://www.mod.uk; T. F. Mills, "Land Forces of Britain, the Empire, and the Commonwealth". http://www.regiments.org; Kenneth Macksey, "First Clash, Combat Close-Up in World War Three," (Royal Tank Regiment), Berkley Books, First edition, March 1, 1988; Carl Schulze, "The British Army of the Rhine (Europa Militaria)", No. 19, Crowood Press, 64 pp., June 1995.

REFERENCES

[1] M. R. Garey, R. L. Graham, and D. S. Johnson, "The complexity of computing Steiner minimal trees," *SIAM Journal on Applied Mathematics*, vol. 32, pp. 835–859, 1977.

[2] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, pp. 374–387, 1996.

[3] P. Klein and R. Ravi, "A nearly best-possible approximation algorithm for node-weighted Steiner trees," *Journal of Algorithms*, vol. 19, pp. 104–115, 1995.

[4] P. Ruiz and A. Gomez-Skarmeta, "Approximating optimal multicast trees in wireless multihop networks," in *Computers and Communications, 2005. ISCC 2005. Proceedings. 10th IEEE Symposium on*, june 2005, pp. 686 – 691.

[5] C.-K. Chau, J. Crowcroft, K.-W. Lee, and S. H. Wong, "Inter-domain routing for mobile ad hoc networks," in *Proc. MobiArch*. New York, NY, USA: ACM, 2008, pp. 61–66.

[6] R. M. Karp, "On the computational complexity of combinatorial problems," *Networks*, vol. 5, pp. 45–68, 1975.

[7] M. R. Garey and D. S. Johnson, "The rectilinear Steiner tree problem is NP-complete," *SIAM Journal on Applied Mathematics*, vol. 32, pp. 826–834, 1977.

[8] R. R. A. Agrawal, P. Klein, "When trees collide: An approximation algorithm for thegeneralized steiner problem on networks," *SIAM Journal on Computing*, vol. 24, no. 3, 1995.

[9] G. Robins and A. Zelikovsky, "Tighter Bounds for Graph Steiner Tree Approximation," *SIAM J. Discret. Math.*, vol. 19, no. 1, pp. 122–134, May 2005.

[10] S. Shi and J. S. Turner, "Routing in overlay multicast networks," in *INFOCOM*, 2002.