Interest-Aware and Bandwidth-Efficient Multicast Group Planning

Selcuk Cevher
M. Umit Uyar
The City College and
Graduate Center of the City University of New York, NY
{scevher,uyar}@ccny.cuny.edu

Mariusz A. Fecko
John Sucec
Sunil Samtani
Applied Research Area
Telcordia Technologies Inc., Piscataway, NJ
{mfecko, jsucec, ssamtani}@research.telcordia.com

Abstract—In multicast group planning in tactical networks (TNs), one of the most important challenges is to maximize the efficient use of network and end-user resources. Hence, multicast groups should be planned cleverly such that the total shared link capacity in a multicast distribution tree is increased and the reception of unwanted traffic at receiver nodes is reduced to a maximum extent. In this paper, we propose a technique to generate a group configuration by estimating the link capacity shared by any two users in a shortest-path multicast tree to improve the branching characteristics such as height and breadth of the tree while reducing the unwanted traffic by considering overlapping user interests. A two-tier information dissemination system architecture for TNs, where a central planner controls many dissemination executives, is a typical application for multicast group planning techniques. We provide simulation results showing that, compared to existing group planning techniques, our approach generates multicast trees with improved branching characteristics while keeping the unwanted traffic within acceptable limits.

Keywords: multicast group planning, shortest-path multicast tree, total shared link capacity

I. INTRODUCTION

In military tactical networks (TNs), multicast group planning is utilized to maximize the efficient use of network and end-user resources including available bandwidth, processing power and battery life. TNs typically handle a large number of information sources requested by a large number of users, each of whom is selectively interested in a subset of these sources. One of the most important challenges in planning the multicast groups is to increase the total shared link capacity while reducing the unwanted traffic at the receiver nodes.

Multicasting enables sending out only one copy of an information to only a subset of users. Although multicasting can reduce bandwidth utilization, it also increases routing state and management overhead. Therefore, multicast is preferable to unicast only if its bandwidth efficiency overcomes the routing state and management overhead [1]. Since the shape of a shortest-path multicast tree (i.e., height and breadth) has a critical impact on multicast bandwidth efficiency, new techniques should be developed to improve the branching characteristics of shortest-path multicast trees. In [2], multicast bandwidth efficiency is shown to increase due to the rise in the number of shared links among receivers as the height of a shortest-path multicast tree grows. Similarly, branching that occurs near the leaves of a multicast tree improves the bandwidth efficiency.

Another potential weakness of multicasting is that certain amount of irrelevant data can be received by multicast users since each user receives all of the flows mapped to the multicast groups to which it subscribes. Especially in TNs, minimizing the amount of unwanted traffic is crucial since the users' resources are very limited. Content filtering is one of the methods proposed to reduce the unwanted traffic reception [3], [4]. However, using filtering while disseminating data to a large receiver set is inefficient since it wastes both network and user resources since the task of determining the interest of a received message is placed on the receiver set [5]. Furthermore, the placement of filtering nodes is a difficult task in mobile wireless networks due to dynamically changing network topology.

Multicast group planning techniques try to find an approximate solution to NP-complete Channelization Problem (CP) defined in [6]. Creating as many multicast groups as the number of information sources is not feasible since it will create an unacceptable multicast state and management overhead. For example, a two-tier information dissemination system architecture for TNs, where a central planner controls many dissemination executives, is a typical application for multicast group planning techniques.

CP is defined as finding an optimal mapping of information flows to multicast groups, and an optimal subscription of users to multicast groups so as to minimize total bandwidth consumption and unwanted traffic. User-Based Merge (UBM), and Flow-Based Merge (FBM) continuously merge two users into the same multicast group based on a pairwise cost function [6]. Both UBM and FBM assumes that the optimal number of multicast groups is pre-determined. On the other hand, localized multicast group update (LMGU) algorithm defined in [7] and [8] generates an adaptive number of multicast groups depending on the underlying network conditions (e.g., end-user tolerance to unwanted traffic).
In this paper, we propose a novel multicast group planning technique by estimating the link capacity shared by any two users in a shortest-path multicast tree to improve branching characteristics and by considering overlapping user interests to reduce the unwanted traffic for each user. We provide simulation results showing that, compared to existing techniques, our approach generates multicast trees with improved branching characteristics while keeping unwanted traffic within acceptable limits. For example, total shared link capacity was observed to increase by more than 10% for dense topologies with 50 and 35 nodes, respectively.

In the next section, we provide background information on CP. We analyze the motivation for our multicast group planning technique and describe our new algorithm in Section III. We provide simulation results for dense and sparse topologies with different number of nodes generated by BRITE topology generator [11] in Section IV. We finally present our conclusions in Section V.

II. BACKGROUND

A. Channelization Problem (CP)

CP in a data dissemination system can be described through three matrices, namely, user-flow interest matrix, flow-to-multicast group mapping matrix, and the subscription matrix, which are defined as follows [6]:

\[ W = (w_{jm})_{N \times M} \]
where \( w_{jm} \) is 1 if receiver \( r_j \) is interested in source \( f_m \), otherwise 0.

\[ X = (x_{im})_{M \times K} \]
where \( x_{im} \) is 1 if flow \( f_i \) is assigned to multicast group \( g_m \), otherwise 0.

\[ Y = (y_{jm})_{N \times K} \]
where \( y_{jm} \) is 1 if receiver \( r_j \) subscribes to multicast group \( g_m \), otherwise 0.

In the above-mentioned equations, \( F = \{ f_1, f_2, ..., f_M \} \) is the set of information sources, \( G = \{ g_1, g_2, ..., g_K \} \) is the set of multicast groups, and \( R = \{ r_1, r_2, ..., r_N \} \) is the set of receivers. Given \( W \) matrix, CP can be defined as finding \( X \) and \( Y \) matrices. Although CP is NP-Complete [6], practical network conditions may not require the optimal solution of CP.

B. Overlap Matrix

An interest profile \( P_i \) of a receiver \( r_i \) specifies the set of information sources that \( r_i \) wants to receive. An \( N \times N \) profile overlap matrix (POM) representing the common information interests among receivers is defined in [9]. The value of each entry \( POM_{ij} \) is the ratio of the number of information sources included in both \( P_i \) and \( P_j \) to the number of information sources contained only in \( P_i \). On the other hand, an \( N \times N \) interest overlap matrix (IOM) is defined in [7] and [8] using a different pairwise cost function. Each entry \( IOM_{ij} \) is defined as:

\[ IOM_{ij} = \frac{|P_i \cap P_j| - |(P_i - P_j) \cup (P_j - P_i)|}{|P_i|} \]
where \( |P_i| \) and \( |P_j| \) represent the common and uncommon information interests between \( r_i \) and \( r_j \), respectively.

For example, given \( W \) in Table I, the interest profiles \( P_1 \) and \( P_2 \) of \( r_1 \) and \( r_2 \) are the sets \{f_2, f_3\} and \{f_1, f_2\}, respectively. \( POM_{12} \) is 1/2 due to \( |P_1 \cap P_2| = 1 \) (i.e., \( P_1 \cap P_2 = \{f_2\} \)) and \( |P_1| = 2 \). However, \( IOM_{12} \) is 1 - 2 = -1 since \( P_1 \cap P_2 = \{f_2\} \) and \( (P_1 - P_2) \cup (P_2 - P_1) = \{f_1, f_3\} \).

III. OUR APPROACH

A. Motivation

In a four-level hierarchical military tactical network (TN), there is typically a direct connection between a layer and the layers below it (Fig. 1) [10]. A Soldier Radio Waveform (SRW) subnetwork is deployed to transmit data, voice, and video flows to the soldiers in a Future Battlefield Network (FBN). In this environment, multicast transmissions are preferred to deliver the ISR (intelligence, reconnaissance and surveillance) data, both for air-to-ground (from UAV sensors) and for ground-to-ground links, and low-bandwidth SRW data from sensor fields. For example, the soldiers on one or more SRW subnets may need to be constantly updated with situation-awareness information gathered from multiple sources, including for example soldiers on another SRW subnet, sensor information from unattended sensors connected through an SRW network, and surveillance data from the UAV sensors. The amount of data from these sources is likely to vary with respect to the volume and update frequency, leading to a dynamic and adaptive multicast dissemination environment. However, multicast deployment in FBNs can be justified only if multicast efficiency overcomes additional routing state and management overhead. Since the shape of a multicast distribution tree has a critical impact on bandwidth efficiency, multicast trees should be created cleverly so that their branching characteristics are improved as much as possible.
In FBNs with highly dynamic mobile nodes interconnected by heterogeneous wireless links, if well-known multicast groups can be pre-planned and pre-loaded into network nodes/hosts, the multicast discovery problem becomes trivial. However, this solution may also typically result in the receipt of unwanted multicast traffic as members of these well-known groups will receive packets from all group sources even though the data from certain sources are not needed. While this may not be a problem in capacity-rich commercial networks, unnecessary multicast traffic could saturate the small-capacity FBN wireless links.

Our proposed solution to CP addresses the improvement in branching characteristics of multicast trees to allow efficient multicast transmissions and the reduction of the receipt of unwanted traffic while ensuring that each user receives all information it requests.

B. Shared Link Matrix

The main benefit of multicasting technique is dependent upon the shape of the multicast distribution tree. If the majority of the paths from the source are not shared by the receivers, multicast is not much more efficient than unicast. In fact, in such a case, multicast may be even less efficient than unicast due to the routing state overhead to implement it. Multicast tree efficiency is determined by two important properties \[2\], namely, height and breadth of the tree. As the height of the multicast tree grows due to newly added receivers near the bottom, the number of shared links among the receivers increases. Early branchings in the tree (i.e., near the root) decreases the multicast efficiency since this will increase the packet duplication and reduce the number of shared links.

A path and a star graph, each with four nodes, are depicted in Figs 2.a and 2.b, respectively, where the black nodes are the receivers and the gray ones are the information sources. A path graph, each receiver node with one incoming edge, has the largest height among all possible trees with the same number of nodes while a star graph, with the information source directly connected to all receivers, has the smallest.

The number of transmission links to be traversed for a single data packet from the information source to reach each receiver in Fig. 2 is shown in Table II. If multicasting is deployed in the topology in Fig. 2.a, three transmission links will be used to deliver a data packet to all receivers since multicasting prevents packet duplication. In the case of unicast, however, the same packet will be repeatedly sent to each receiver, utilizing a total of six transmission links. On the other hand, multicast and unicast schemes use the same number of transmission links in the topology in Fig. 2.b due to the branching at the information source. This difference in the relative multicasting efficiency is an indication of how much the height of the distribution tree impacts the performance of multicasting.

Let us define a matrix to store the number of shared links among all user pairs in a multicast distribution tree.

**Definition 1:** Let \(r_i\) and \(r_j\) \((i, j \in \mathbb{N}; i < j)\) be two receivers in a multicast distribution tree. The number of links shared by \(r_i\) and \(r_j\) is defined as the size of the largest subpath rooted at the information source which is traversed by both of the shortest paths from the information source to \(r_i\) and \(r_j\).

**Definition 2:** Shared link matrix \(S\) is defined as the symmetric square matrix \((s_{ij})_{N \times N}\) where \(N\) is the number of receivers in the multicast distribution tree, and each entry \(s_{ij}\) contains the number of links shared by the receivers \(r_i\) and \(r_j\) for \(i \neq j\).

![Figure 2](image-url)

Fig. 2. (a) A Path graph with 4 nodes (b) A star graph with 4 nodes

Table III shows the shared link matrices for Figs 2.a and 2.b. Any element \(s_{ij}\), where \(i = j\) or \(i > j\), is not included in the computation since \((1)\) the number of links that a receiver \(r_i\) shares with itself is already the length of the shortest path from the information source to \(r_i\), and \((2)\) matrix \(S\) is symmetric. Such \(s_{ij}\) \((i = j,\ or i > j)\) elements are denoted by \(x\) in Table III. As can be seen in Table III, a larger number of transmission links are shared among the receivers in the multicast distribution tree if the height of the tree increases.

**C. Our Algorithm (IBMP)**

IBMP aims to estimate the number of links which will be shared by each pair of receivers in a multicast tree. It merges the user pairs who will share a relatively larger number of transmission links with a high probability into the same multicast group. For this purpose, it sequentially finds the shortest-path trees (SPTs) for a given network topology where the root of the tree is the information source. The possible
number of transmission links in a multicast tree which will be shared by a receiver pair is determined by calculating the mean of the number of shared links in each individual SPT of each information source.

IBMP consists of two tasks, namely, finding the estimated number of shared links for each receiver pair in a multicast tree (i.e., $\overline{S}$ matrix) and combining $\overline{S}$ with IOM matrix. The algorithms for these tasks are shown in Fig. 3 and Fig. 4, respectively. The current topology $G$, the set of information sources $F$ and the set of receivers $R$ are the inputs for the algorithm in Fig. 3 where $|F|$ denotes the total number of SPTs. Function called $findSP(G, f_i)$ employs the Dijkstra's algorithm to find the SPT starting from the the information source $f_i$. Function called $findS(SPT)$ computes the $S$ matrix for $SPT$ found by $findSP$. Using the two inner loops in Fig. 3, the estimated number of shared links for each receiver pair in the multicast tree is computed as the mean number of shared links in $|F|$ shortest-path trees.

IBMP presents a tradeoff between the unwanted traffic and the number of shared links for each pair of receivers $r_i$ and $r_j$ ($i, j \in N; i < j$) using the simple function $f$ to combine each entry $s_{ij}$ in $\overline{S}$ with the corresponding element $m_{ij}$ in IOM:

$$f(s_{ij}, m_{ij}) = \lambda \cdot s_{ij} + (1 - \lambda) \cdot m_{ij}$$

where $\lambda$ (0 $\leq$ $\lambda$ $\leq$ 1) determines the contribution of each entry in $\overline{S}$ to the corresponding entry in the combined matrix $S^c_{ij}$. The algorithm for combining $\overline{S}$ with IOM is presented in Fig. 4.

To evaluate the performance of our IBMP algorithm, we utilized a modified version of localized multicast group update (LMGU) algorithm described in [7]. LMGU algorithm uses IOM to merge a user pair with maximum merging benefit into the same multicast group. We added a new input parameter to LMGU algorithm enabling us to execute LMGU on $S^c$ rather than on IOM.

Let us describe the computation of $\overline{S}$ matrix using the example topology with seven nodes shown in Fig. 5 where dotted and solid circles represent information sources and receivers, respectively. For simplicity, each node in Fig. 5 is assumed to have a single functionality, namely, either an information source or a receiver. The numbers over the edges denote the link capacities. IBMP sequentially finds the SPTs of the entire topology of Fig. 5 for the informaton sources numbered as 2, 5 and 7. These SPTs are shown in Figs 6.a, 6.b and 6.c, respectively. Table IV shows the $S$ matrices computed for each SPT in Fig. 6. For example, nodes 1 and 3 in Fig. 6.a share no transmission links in Table IV.a since the shortest paths starting at node 2 to each of these nodes do not intersect. On the other hand, the shortest path to node 6 contains the shortest path to node 1 resulting a shared transmission link with a capacity of 10.

Table V presents the $\overline{S}$ matrix computed based on the $S$ matrices in Table IV. Each entry $s_{ij}$ in $\overline{S}$ is calculated as the mean value of $s_{ij}$ elements in $S$ matrices shown in Table IV. For example, the estimated shared link capacity between nodes 1 and 6 in Table V is $(10 + 2 + 15)/3 = 9$ where 3 represents the total number of information sources.

In conclusion, merging the nodes 1 and 6 in the example topology of Fig. 5 into the same multicast group will result in
a larger amount of shared link capacity with a high probability since 9 is the maximum entry in $S$ presented in Table V.

<table>
<thead>
<tr>
<th>node 1</th>
<th>node 3</th>
<th>node 4</th>
<th>node 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>$x$</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>$x$</td>
<td>8.333</td>
<td>0.666</td>
<td>9</td>
</tr>
<tr>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>0.666</td>
</tr>
<tr>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
</tbody>
</table>

TABLE IV
S matrices computed for the example topology in Fig. 5.

<table>
<thead>
<tr>
<th>node 1</th>
<th>node 3</th>
<th>node 4</th>
<th>node 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>$x$</td>
<td>$x$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x$</td>
<td>$x$</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
</tbody>
</table>

TABLE V
$S$ matrix computed based on the $S$ matrices in Table IV.

IV. Simulation Results

We used BRITE topology generator [11] in Bottom-up mode for our simulations. To create realistic topologies, we relied on the experiments performed in [12] and [13] to select the BRITE parameters listed in Table VI. If $bw$ distribution is set to constant, all link capacities have identical values, namely, minimum $bw = 100$. Otherwise, a uniform distribution ranging from 100 to 1024 is adopted. Sparse and dense topologies with 20, 35 and 50 nodes were generated by adjusting $p$ (add) and beta parameters which affect the probability of interconnection between any two nodes. To create sparse topologies, we used the small value of 0.01 for both $p$ (add) and beta. The values of 0.44 and 0.63 were utilized to generate dense topologies. For each generated topology, we set $|F| = |V|/4$ where $|F|$ and $|V|$ denote the total number of information sources and the total number of nodes in the topology, respectively (i.e., $|N| = |V| - |F|$ where $|N|$ is the total number of receivers). The transmission rate of each information source $f_i$ was randomly selected as either 10 or 100.

We evaluated the performance of our IBMP algorithm in terms of two different metrics, namely, unwanted traffic and total shared link capacity. For this purpose, we sequentially executed LMGU algorithm on $S^c$ and $IOM$ matrices for different values of $0 \leq \lambda \leq 1$, and compared unwanted traffic and total shared link capacity in various source-based multicast trees with 4, 6 and 8 subscribers.

The results obtained from our simulations are shown in Figs. 7 through 16. The information presented in each of these figures was obtained as the mean of 10 different topologies. The title of each figure specifies the size and density (i.e., $p$ (add) and beta) of the topology, and the link capacity distribution (i.e., uniform or constant) used in the corresponding simulation. The $x$-axis of each figure (i.e., $lambda$) refers to $\lambda$ in Eq. 1 while $y$-axis represents the total shared link capacity or unwanted traffic per user. Each figure legend demonstrates the number of multicast groups (i.e., 4, 6, or 8) and the type of the LMGU input matrix (i.e., active for $S^c$, and inactive for $IOM$) used in the respective simulation.

Figs. 7 through 10 show the simulation results obtained for sparse/dense topologies with 20 nodes where link capacities have a uniform distribution. For four multicast groups, the total shared link capacity was observed to increase by more than $100$ for $\lambda = 1$ (Fig. 9) when LMGU algorithm was executed on $S^c$ matrix. The total shared link capacity remained the same regardless of $\lambda$ when LMGU algorithm utilized $IOM$ matrix since $IOM$ does not reflect any consideration for the shared link capacity. As expected, we obtained similar curves for unwanted traffic per user (Fig. 10) since the contribution of the user interests (i.e., $1 - \lambda$) to each entry in the combined matrix $S^c$ in Eq. 1 decreases as $\lambda$ grows. In general, the decision on the value of $\lambda$ depends on the specific network conditions (e.g., network bandwidth and user tolerance to unwanted traffic). In sparse and dense topologies with 35 nodes, for 8 multicast groups, the total shared link capacities increase by $100$ for $\lambda = 0.3$ for sparse topologies (Fig. 11), and by more than $100$ for $\lambda = 0.75$ for dense topologies (Fig. 12). Interesting results were observed for sparse and dense topologies with 50 nodes shown in Figs. 13 through 16. For example, as opposed to the expected behaviour, the total shared link capacity increased by $18$ for 4 multicast groups and $\lambda = 0.75$ (Fig. 15) while no additional unwanted traffic per user was generated (Fig. 16).

V. Conclusions

In this paper, we propose a multicast group planning technique to generate a group configuration by estimating the transmission link capacity shared by any two users in a
Fig. 7. Sparse topology with 20 nodes, uniformly distributed link capacities

Fig. 8. Sparse topology with 20 nodes, uniformly distributed link capacities

Fig. 9. Dense topology with 20 nodes, uniformly distributed link capacities

Fig. 10. Dense topology with 20 nodes, uniformly distributed link capacities

Fig. 11. Sparse topology with 35 nodes, uniformly distributed link capacities

Fig. 12. Dense topology with 35 nodes, identical link capacities
shortest-path multicast tree to improve the branching characteristics (i.e., height and breadth) of the tree while reducing the unwanted traffic per user by considering overlapping user interests. We provide simulation results showing that, compared to existing group planning techniques, our approach generates multicast trees with improved branching characteristics while keeping the unwanted traffic within acceptable limits. For example, total shared link capacity was observed to increase by \(18\%\) and more than \(100\%\) for dense topologies with 50 and 35 nodes, respectively.

REFERENCES


1The 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 Army Research Lab or the U.S. Government.