Content Distribution by Multiple Multicast Trees and Intersession Cooperation: Optimal Algorithms and Approximations

Xiaoying Zheng, Chunglae Cho and Ye Xia

Abstract— **The paper addresses the problem of massive content distribution in a network where multiple sessions coexist. In more traditional approaches, the sessions form separate overlay networks and operate independently from each other. In this case, some sessions may suffer from insufficient resources (e.g., aggregate upload bandwidth) even though other sessions have excessive resources. To cope with this problem, we consider the *universal swarming* approach, which allows multiple sessions to cooperate with each other by forming a shared overlay network. We formulate the problem of finding the optimal resource allocation to maximize the sum of the session utilities under the network capacity constraints. The solution turns out to be optimal sharing of multiple minimum-cost multicast trees. We present a subgradient algorithm and prove that, although the algorithm uses a single multicast tree per session at each iteration and hence does not converge in the conventional sense, it converges to the optimal solution in the time-average sense. The solution involves an NP-hard subproblem of finding a minimum-cost Steiner tree. We cope with this difficulty by using a column generation method, which reduces the number of Steiner-tree computations. Furthermore, we allow the use of approximate solutions to the Steiner-tree subproblem. We show that the approximation ratio to the overall problem turns out to be no more than that to the Steiner-tree subproblem. Simulation results demonstrate that universal swarming improves the performance of resource-poor sessions with negligible impact to resource-rich sessions.**

I. Introduction

The Internet is being applied to transfer content on a more and more massive scale. While many content distribution techniques have been introduced, most of the recently introductions are based on the *swarming* technique, such as FastReplica [1], Bullet [2], [3], Chunkcast [4], BitTorrent [5], and CoBlitz [6]. In a swarming session, the file to be distributed is broken into many chunks at the source node, which are then spread out to the receivers; the receivers will then help each other with the retrieval of the missing chunks. By taking advantage of the resources of the receivers, swarming dramatically improves the distribution efficiency (e.g., average downloading rate, completion time) compared to the traditional client-server-based approach.

The swarming technique was originally created by the end-user communities for peer-to-peer (P2P) file sharing. The subject of this paper is how to apply swarming to infrastructure-based content distribution and make the distribution more efficient. Compared with the dynamic end-user file-sharing situation, infrastructure networks and content servers are much more stable. In this setting, we will see that it is beneficial to view swarming as distribution over multiple multicast trees. This view allows us to pose the question of how to optimally distribute the content. (See [7].)

The specific problem addressed in this paper is how to conduct content distribution more efficiently in a network where multiple distribution sessions coexist. A distribution *session* consists of a file to be distributed, one or more sources and all the nodes who wish to receive the file, i.e., the receivers. Different sessions may have heterogenous resource capacities, such as the source upload bandwidth, receiver download bandwidth, or aggregate upload bandwidth. For instance, there may exist some sessions with excessive aggregate upload bandwidth because their throughput bottleneck is at the source upload bandwidth, the receiver download bandwidth, or the internal network; at the same time, there may exist some other sessions whose throughput bottleneck is at their aggregate upload bandwidth. In the traditional swarming approach, the sessions operate independently by each forming a separate overlay network; this will be called *separate swarming*, which does not provide the opportunity for the resource-poor sessions to use the surplus resources of the resource-rich sessions. However, if we conduct *universal swarming*, that is, we combine multiple sessions together into a single "super session" on a shared overlay network and allow them to share each other's resources, the distribution efficiency of the resource-poor sessions can improve greatly with negligible impact on the resource-rich sessions. The paper examines algorithms and theoretical issues related to universal swarming.

We first need to establish the equivalence of swarming and distribution over multiple multicast trees. More details are discussed in [7]. Consider following the distribution paths of each individual file chunk. With some thought, it can be seen that the chunk is distributed over a tree rooted at the source and covering all the receivers of the file.[1] Since the chunks travel down different trees, the distribution of each file involves multiple multicast trees. In universal swarming, a distribution tree not only includes all the receivers interested in downloading the file but may also contain nodes that are not interested in the file; the latter will be called *out-of-session nodes*. Thus, each distribution tree for a session is a *Steiner* tree rooted at the source where all the receivers are terminal nodes and the out-of-session nodes on the tree are Steiner nodes.

With the tree-based model, the optimal distribution prob-

Xiaoying Zheng, Chunglae Cho and Ye Xia are with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611. Email: {xiazheng, ccho, yx1}@cise.ufl.edu.

[1] For ease of discussion, we assume that each file has a single source. This is without loss of generality.

lem can be formulated as finding an optimal rate allocation on the multiple multicast trees so that it achieves the optimal performance objective. A version of this problem was addressed in [7] in the context of separate swarming. The rate-allocation problem in universal swarming, which this paper concerns, is substantially more difficult. The main reason is that, by the optimality condition, an optimal solution typically uses only the minimum-cost trees to distribute the file chunks. Hence, an optimal universal swarming algorithm usually involves an NP-hard subproblem of finding a minimum-cost (min-cost) Steiner tree. How to cope with this issue is one of the main themes in this paper.

We present two solution approaches, which can be used in combination. First, we introduce into our rate-allocation algorithm a column generation method, which can reduce the number of times the min-cost Steiner-tree is computed. Second, we allow the use of approximate solutions to the Steiner-tree subproblem. Some approximate solutions to the Steiner-tree problem in directed graphs can be found in [8]–[10]. Importantly, we show that the approximation ratio to the overall rate-allocation problem turns out to be no more than that to the Steiner-tree subproblem.

The overall rate-allocation algorithm that we will present is a subgradient algorithm. It has the characteristic of assigning positive rate to a single multicast tree per session at each iteration; the rate assigned to the tree is computed based on the link prices at the iteration. We can show that even though the assigned rates in each iteration usually exceed the capacities of some links, the time-average rates satisfy the link capacity constraints, and eventually the rate allocation to each session converges to the optimum (provided the Steiner-tree subproblem is solved optimally.) It is worth pointing out that other optimization algorithms may also be used here instead of the subgradient algorithm.

We now briefly discuss additional related work. A heuristic centralized algorithm for the multicast tree packing problem is proposed in [11]. Jansen et. al. present a centralized approximation algorithm for the multicast congestion control problem in [12]. [13]–[15] apply the network coding technique to achieve the multicast capacity; part of their solution techniques is similar to ours. A survey of optimization problems in multicast routing can be found in [16]. [17]–[19] model and analyze the peer-assisted file distribution system. The multipath routing problem has been studied in [20]–[23].

The paper is organized as follows. The formal problem description is given in Section II. The subgradient algorithm and its convergence results are given in Section III. In Section IV, we present the column generation approach, combine it with the subgradient algorithm, and study the performance bound when approximation algorithms are applied to the min-cost tree subproblem. For brevity, the proofs are omitted, which can be found in an extended version of this paper [24]. We show some simulation results about our approach in Section V. The conclusion is drawn in Section VI.

II. Problem Description

Let the network be represented by a directed graph $G = (V, E)$, where V is the set of nodes and E is the set of links. For each link $e \in E$, let c_e be its capacity, where $c_e > 0$. A *multicast session* is associated with a file and consists of the source node and all the receivers of the file. Let s denote a session or the source of a session interchangeably. In a session s, the data traffic is routed along multiple multicast trees, each rooted at the source s and covering all the receivers. A multicast tree is a *Steiner* tree; it may contain nodes not in the session, which are called Steiner nodes. Let the set of all allowed multicast trees for session s be denoted by T_s. Throughout the paper, we assume T_s contains all possible multicast trees unless specified otherwise. Let S be the set of all multicast sessions, and let $T = \cup_{s \in S} T_s$. Then, T is the collection of all multicast trees for all sessions. The multicast trees can be indexed in an arbitrary order as $t_1, t_2, \cdots, t_{|T|}$, where $|\cdot|$ is the cardinality of a set. Though $|T|$ is finite, it is usually very large. Let x_s be the flow rate of session $s \in S$ and y_t be the flow rate of a multicast tree t. We have $x_s = \sum_{t \in T_s} y_t$.

Each session $s \in S$ is associated with a utility function $U_s(x_s), 0 \leq m_s \leq x_s \leq M_s$. The assumption on the utility functions is, for every $s \in S$,

- $A1$: U_s is well-defined (real-valued), non-decreasing, strictly concave on $[m_s, M_s]$, and twice continuously differentiable on (m_s, M_s).

The problem is to find the optimal resource (i.e., session rate and multicast-tree rate) allocation to maximize the sum of session utilities under the capacity constraints and session rate constraints. We call the optimization problem the master problem (MP), which is as follows.

$$\begin{array}{llll}
\text{MP:} & \max\ f(x,y) = \sum_{s \in S} U_s(x_s) & & (1) \\
\text{s.t.} & x_s = \sum_{t \in T_s} y_t, & \forall s \in S & \\
 & \sum_{t \in T: e \in t} y_t \leq c_e, & \forall e \in E & (2) \\
 & m_s \leq x_s \leq M_s, & \forall s \in S & \\
 & y_t \geq 0, & \forall t \in T. &
\end{array}$$

We make an assumption about the MP, which is almost always satisfied in practice.

- $A2$: There exists a feasible solution $(\bar{x}, \bar{y})$ such that $m_s \leq \bar{x}_s \leq M_s$ for every session $s \in S$, $f(\bar{x}, \bar{y}) > -\infty$ and (2) holds with strict inequality at $(\bar{x}, \bar{y})$.

Note that $f(x, y)$ is strictly concave on x, but linear on y.

Let λ_e be the Lagrangian multiplier associated with the constraint (2). The Lagrangian function of (1) is

$$\begin{aligned}
L(x,y,\lambda) &= \sum_{s \in S} U_s(x_s) + \sum_{e \in E} \lambda_e (c_e - \sum_{t \in T: e \in t} y_t) \\
&= \sum_{s \in S} (U_s(x_s) - \sum_{t \in T_s} y_t \sum_{e \in t} \lambda_e) + \sum_{e \in E} \lambda_e c_e. \quad (3)
\end{aligned}$$

The dual function is

$$\begin{aligned}\theta(\lambda) = \quad & \max \quad L(x,y,\lambda) && (4)\\ \text{s.t.} \quad & x_s = \textstyle\sum_{t\in T_s} y_t, \quad \forall s \in S\\ & m_s \le x_s \le M_s, \quad \forall s \in S\\ & y_t \ge 0, \qquad \forall t \in T.\end{aligned}$$

Now the dual problem of (1) is

$$\begin{aligned}\text{Dual:} \quad & \min \quad \theta(\lambda) && (5)\\ \text{s.t.} \quad & \lambda \ge 0.\end{aligned}$$

III. A Distributed Algorithm

In this section, we illustrate how the problem MP in (1) can be solved by a distributed subgradient algorithm. In Section IV, we will combine this algorithm with a column generation method and derive a family of algorithms.

A. Subgradient Algorithm

The dual problem (5) can be solved by a standard subgradient method as in Algorithm 1, where $\delta_e(k)$ is a positive scalar step size, $[\cdot]_+$ and $[\cdot]_{m_s}^{M_s}$ denote the projection onto the non-negative domain and on the interval of $[m_s, M_s]$, respectively [25] [26]. There are two step size rules:

- Rule I (Constant step size): $\delta_e(k) = \delta > 0$, for all time $k \ge K$ for some finite K.
- Rule II (Diminishing step size): $\delta_e(k) \le \delta_e(k-1)$ for all time $k \ge K$, for some finite K; $\lim_{k\to\infty} \delta_e(k) = 0$; and $\lim_{k\to\infty} \sum_{u=0}^{k} \delta_e(u) = \infty$.

At the update (9) and (10) in Algorithm 1, we need to compute a min-cost Steiner tree. Under any fixed dual cost vector $\lambda \ge 0$ and for any session $s \in S$, let us denote a min-cost Steiner tree by

$$t(s,\lambda) = \text{argmin}_{t\in T_s}\{\sum_{e\in t} \lambda_e\}, \tag{6}$$

where a tie is broken arbitrarily. Because (6) is an optimization problem over all allowed trees, we call (6) a *global min-cost tree problem*, and the achieved minimum cost the *global minimum tree cost*. We denote this global minimum tree cost under a fixed $\lambda \ge 0$ by

$$\gamma(s,\lambda) = \sum_{e\in t(s,\lambda)} \lambda_e. \tag{7}$$

Algorithm 1 Subgradient Algorithm

$$\lambda_e(k+1) = [\lambda_e(k) - \delta_e(k)(c_e - \sum_{t\in T: e\in t} y_t(k))]_+, \forall e \in E \tag{8}$$

$$x_s(k+1) = [(U_s')^{-1}(\gamma(s,\lambda(k+1)))]_{m_s}^{M_s}, \forall s \in S \tag{9}$$

$$y_t(k+1) = \begin{cases} x_s(k+1) & \text{if } t = t(s,\lambda(k+1));\\ 0 & \text{otherwise,}\end{cases} \quad \forall t \in T. \tag{10}$$

Remark: Algorithm 1 is a distributed algorithm. In order to compute the tree cost, each link e can independently compute its dual cost λ_e based on the local aggregate rate passing through the link. Then, the tree cost can be accumulated by the source s based on the link cost values along the tree. We will address the issue of finding the min-cost tree in Section IV, which is an NP-hard problem. Other than that, the subgradient algorithm can be completely decentralized.

B. Convergence Results

Let $\Lambda^* = \{\lambda \ge 0 : \theta(\lambda) = \min_{\lambda\ge 0} \theta(\lambda)\}$ be the set of optimal dual variables. Let (x^*, y^*, λ^*) denote one of the optimal primal-dual solutions. Note that x^* is unique, but y^* and λ^* may not be.

Theorem 1: Let $d(\lambda,\Lambda^*) = \min_{\lambda^*\in\Lambda^*} ||\lambda - \lambda^*||$. For any $\epsilon > 0$, under either the step size rule I or II, there exist a sequence of step sizes $\{\delta(k)\}$ and a sufficiently large $K_0 < \infty$ such that, with any initial $\lambda(0) \ge 0$, for all $k \ge K_0$, $d(\lambda(k),\Lambda^*) < \epsilon$ and $||x(k) - x^*|| < \epsilon$ [27].

We now discuss the convergence of the tree rate vector $y(k)$. The difficulty of proving the convergence of $y(k)$ arises from the linearity of the Lagrangian function in (3) on the vector y, and there is no standard result about the convergence of $y(k)$. In fact, the tree rate vector $y(k)$ does not converge in the normal sense [28]. From the update (10), we see that each source only uses one tree (i.e., assigns a positive rate) each time and shifts flow from one tree to another from time to time. We further notice that, by pushing the session flow onto only one tree at a time, the link capacity constraints are often violated. This means that the rate allocation on each time slot may not even be feasible. In Theorem 2, we will show that the tree rate converges in the time-average sense.

Let H denote the $|E| \times |T|$ link-tree incidence matrix where $[H]_{et} = 1$ if $e \in t$; otherwise, $[H]_{et} = 0$. Let A denote the $|S| \times |T|$ session-tree incidence matrix where $[A]_{st} = 1$ if $t \in T_s$; otherwise, $[A]_{st} = 0$. For an arbitrary k_0, where $k_0 \ge 0$, let us define a sequence $\{\bar{y}(k)\}_{k\ge k_0}$, where

$$\bar{y}(k) = \frac{\sum_{u=k_0}^{k} y(u)}{k - k_0 + 1}. \tag{11}$$

For any $\epsilon > 0$, let us define $\mathcal{Y}^*(\epsilon) = \{y \ge 0 : Hy \le c, ||Ay - x^*|| \le \epsilon\}$. When $\epsilon = 0$, $\mathcal{Y}^*(\epsilon) = \mathcal{Y}^* = \{y \ge 0 : Hy \le c, Ay = x^*\}$, which is the set of optimal tree rates.

Theorem 2: For any $\epsilon > 0$, with any initial $\lambda(0) \ge 0$, every limit point of the sequence $\{\bar{y}(k)\}$ is in the set $\mathcal{Y}^*(\epsilon)$.

Remark: By Theorem 2, the time average of the tree rate vectors, $\bar{y}(k)$, converges to the optimal set. Theorem 2 holds under both the step size rule I and II.

IV. Column Generation Method With Imperfect Global Min-cost Tree Scheduling

In Section III, we develop a distributed algorithm to solve the master problem (1), if the min-cost Steiner tree subproblem (6) can be solved precisely in a distributed fashion. However, the subproblem (6) is NP-hard [29]. The column generation method can be introduced to reduce the number of times that the min-cost Steiner tree subproblem is invoked. We also consider applying imperfect tree scheduling, which are approximate or heuristic, sub-optimal solutions to the Steiner tree subproblem. This column generation method with approximation was first proposed in [30] to solve the problem of wireless link scheduling.

A. Column Generation Method

The main idea of column generation is to start with a subset of the tree set T and bring in new trees only when needed. Consider a subset of T containing only a small number of trees, i.e., $T^{(q)} = \{t_i \in T : \forall i = 1, \cdots, q\}$. We make sure that $T^{(q)}$ contains at least one tree for each source s. Denote $T_s^{(q)}$ the subset of trees in $T^{(q)}$ that are rooted at source s, i.e., $T_s^{(q)} = \{t : t \in T^{(q)} \cap T_s\}$. We can formulate a restricted master problem (RMP) by replacing T with $T^{(q)}$ in the MP (1), and this will be called the q^{th}-RMP.

The value of q is usually small and the trees in the set $T^{(q)}$ can be examined one-by-one. The Lagrangian function, the dual function, and the dual problem of the q^{th}-RMP can be formulated similarly as in (3), (4), and (5), where the set T is replaced by the set $T^{(q)}$.

The q^{th}-RMP is more restricted than the MP. Thus, any optimal solution to the q^{th}-RMP is feasible to the MP and serves as a lower bound of the optimal value of the MP. By gradually introducing more trees (columns) into $T^{(q)}$ and expanding the subset $T^{(q)}$, we will improve the lower bound of the MP [31]–[33].

B. Apply the Subgradient Algorithm to the RMP

The distributed subgradient algorithm can be used to solve the q^{th}-RMP. Here, we define the following problem of finding the min-cost tree $t^{(q)}(s, \lambda)$ under the link cost vector $\lambda \geq 0$.

$$t^{(q)}(s, \lambda) = \text{argmin}_{t \in T_s^{(q)}} \{\sum_{e \in t} \lambda_e\}, \qquad (12)$$

The optimization is taken over the $|T_s^{(q)}|$ currently known trees. The problem in (12) is called the *local min-cost tree problem*, and the achieved minimum cost is called the *local minimum tree cost*. We denote this local minimum cost under $\lambda \geq 0$ by

$$\gamma^{(q)}(s, \lambda) = \sum_{e \in t^{(q)}(s,\lambda)} \lambda_e. \qquad (13)$$

If there is more than one tree achieving the local minimum cost, the tie is broken arbitrarily.

C. Introduce One More Tree (Column)

Now the question is how to check whether the optimum of the q^{th}-RMP is optimal for the MP, and if not, how to introduce a new column (tree). It turns out there is an easy way to do both. Let $(\bar{x}^{(q)}, \bar{y}^{(q)}, \bar{\lambda}^{(q)})$ denote one of the optimal primal-dual solutions of the q^{th}-RMP.

Lemma 3: $(\bar{x}^{(q)}, \bar{y}^{(q)}, \bar{\lambda}^{(q)})$ is optimal to the MP if and only if $h_s(\gamma(s, \bar{y}^{(q)})) = h_s(\gamma^{(q)}(s, \bar{y}^{(q)}))$, for all $s \in S$, where

$$h_s(w) = U_s([(U_s')^{-1}(w)]_{m_s}^{M_s}) - [(U_s')^{-1}(w)]_{m_s}^{M_s} \cdot w, \ w \geq 0.$$

From Lemma 3, a sufficient condition for optimality is that the local minimum tree cost is equal to the global minimum tree cost (i.e., $\gamma(s, \bar{\lambda}^{(q)}) = \gamma^{(q)}(s, \bar{\lambda}^{(q)})$), which implies $h_s(\gamma(s, \bar{\lambda}^{(q)})) = h_s(\gamma^{(q)}(s, \bar{\lambda}^{(q)}))$). We state the rule of introducing a new column in the following.

Fact 4: Any tree achieving a cost less than the local minimum tree cost could enter the subset $T^{(q)}$ in the RMP. The tree achieving the global minimum tree cost is one possible candidate and is often preferred [30].

D. Column Generation by Imperfect Global Tree scheduling

The min-cost Steiner tree subproblem (6) is NP-hard, which makes the step of column generation very difficult. We now consider approximation algorithms to this subproblem. We may solve it approximately, and this is referred as *imperfect global tree scheduling*.[2]

Suppose we are able to solve (6) with an approximation ratio $\rho \geq 1$, i.e.,

$$\gamma(s, \lambda) \leq \gamma_\rho(s, \lambda) \leq \rho\gamma(s, \lambda), \qquad (14)$$

where $\gamma_\rho(s, \lambda)$ is the cost of the tree given by the approximate solution.

1) A ρ-Approximation Approach: We develop a column generation method with imperfect global min-cost tree scheduling as follows. Later, we will show a guaranteed performance bound of this approach. Algorithm 2 in fact describes a whole class of algorithms representing different performance, convergence speed and complex tradeoffs. More detailed comments about the property of this class of algorithms can be found in [30].

Algorithm 2 Column Generation with Imperfect Global Tree Scheduling

```
• Initialize: Start with a collection of $T^{(q)}$ trees. (Assume
  Assumption $A2$ holds for the $q^{th}$-RMP.)
• Step 1: Run the subgradient algorithm (8)-(10) for
  several (a finite number) times on the $q^{th}$-RMP.
• Step 2: For each source $s$, solve the global min-cost
  tree problem (6) with an approximation ratio $\rho$ under
  the current dual cost $\lambda$.
    – If the tree corresponding to the approximate solution of (6) is already in the current collection of
      trees, do nothing;
    – Otherwise, introduce this tree into the current collection of trees, and increase $q$ by $1$.
  Go to Step $1$.
```

2) Convergence with Imperfect Global Tree Scheduling:

Theorem 5: There exists a q, $1 \leq q \leq |T|$, such that Algorithm 2 converges to one optimal primal-dual solution of this particular q^{th}-RMP, i.e., $(\bar{x}^{(q)}, \bar{\lambda}^{(q)})$. Furthermore, after Algorithm 2 converges to $(\bar{x}^{(q)}, \bar{\lambda}^{(q)})$, $\gamma_\rho(s, \bar{\lambda}^{(q)}) = \gamma^{(q)}(s, \bar{\lambda}^{(q)})$ for any source $s \in S$.

[2] Note that the local min-cost tree problem (12) can be easily solved precisely since the number of extreme points (i.e., candidate trees) of $T^{(q)}$ is usually small, and hence, enumerable.

3) *Performance Bound under Imperfect Tree Scheduling:*

Theorem 5 says that the column generation method with imperfect global tree scheduling converges to a sub-optimum of the MP. We will prove that the performance of this sub-optimum is bounded. We make the assumptions $A3$ and $A4$.

- $A3$: For any source $s \in S$, $m_s \geq 0$ is sufficiently small such that, if the column generation method with imperfect global tree scheduling converges to $(\bar{x}^{(q)}, \bar{\lambda}^{(q)})$ on the q^{th}-RMP, then $\bar{x}_s^{(q)} > m_s$.
- $A4$: $U_s(m_s) - m_s \cdot U_s'(m_s) \geq 0, \forall s \in S$.

Theorem 6 (Bound of Imperfect Global Tree Scheduling): Under the additional assumptions $A3$ and $A4$, if the column generation method with imperfect global tree scheduling converges to $(\bar{x}^{(q)}, \bar{\lambda}^{(q)})$ on the q^{th}-RMP, we have

$$\theta^{(q)}(\bar{\lambda}^{(q)}) \leq \sum_{s \in S} U_s(x_s^*) \leq \theta(\rho\bar{\lambda}^{(q)}) \leq \rho\theta^{(q)}(\bar{\lambda}^{(q)}). \quad (15)$$

Since the strong duality holds on the q^{th}-RMP, $\sum_{s \in S} U_s(\bar{x}_s^{(q)}) = \theta^{(q)}(\bar{\lambda}^{(q)})$, we have the following.

Corollary 7 (ρ-Approximation Solution to the MP): Under the additional assumptions $A3$ and $A4$, we have

$$\sum_{s \in S} U_s(\bar{x}_s^{(q)}) \leq \sum_{s \in S} U_s(x_s^*) \leq \rho \sum_{s \in S} U_s(\bar{x}_s^{(q)}). \quad (16)$$

If $\rho = 1.0$, (16) holds with equality, then Algorithm 2 is the column generation method with perfect global min-cost tree scheduling, and it converges to one optimum of MP.

Corollary 7 says that the column generation method with imperfect global tree scheduling converges to a sub-optimum of the MP and achieves an approximation ratio no more than the approximate solution to the global min-cost tree problem.

Remark: Possible utility functions include $U_s(x_s) = w_s \ln(x_s + e)$ and $U_s(x_s) = \frac{w_s}{1-\beta} x_s^{1-\beta}$, where $0 < \beta < 1$ and $w_s > 0$.

V. Illustrative examples

In this section, we give illustrative examples showing the effect of universal swarming and the performance of our algorithms. We show that the subgradient algorithm achieves the optimum in the time-average sense.

We test our algorithms in various scenarios by varying the sizes of the resource-rich and resource-poor sessions and the locations of bandwidth bottleneck. We have nine test cases (profiles) where we assume the internal network has large capacity so that it cannot be the bottleneck; therefore, the bottleneck lies on the access links. In each of the profiles $A1$, $A2$ and $A3$, there is a large resource-rich session (RRS) and a small resource-poor session (RPS); in each of the profiles $B1$, $B2$ and $B3$, there is an RRS and an equal-sized RPS; and in each of the profiles $C1$, $C2$ and $C3$, there is a small RRS and a large RPS. Each large session contains 90 receivers; each small session contains 10 receivers; and each medium session contains 50 receivers. Each session has a single source. We also vary the bottleneck of the sessions so that we can examine how intersession cooperation affects the rate allocation in each case. In profiles $A1$, $B1$ and $C1$, the bottleneck of the RRS is at the download links; in profile $A2$, $B2$ and $C2$, the bottleneck of the RRS is at the upload link of its source; and in profile $A3$, $B3$ and $C3$, the RRS is bottlenecked by its aggregate upload bandwidth. In all cases, the RPS is bottlenecked at its aggregate upload bandwidth. Note that if the bottleneck of the RPS is at its source upload link or the receiver download links, then there is no way to improve its session rate.

TABLE I

COMPARISON OF RATE ALLOCATION BETWEEN SEPARATE SWARMING AND UNIVERSAL SWARMING

Test cases		Link bandwidth			Rate allocation	
Profile	Session	u_s	u_i	d_i	Separate	Universal
A1	Large RRS	640	360	360	360	329.5
	Small RPS	640	36	360	100	359.7
A2	Large RRS	280	360	360	280	280
	Small RPS	280	36	360	64	280
A3	Large RRS	640	200	360	207	170.2
	Small RPS	640	20	360	84	360
B1	Medium RRS	640	360	360	360	205.6
	Medium RPS	640	36	360	48.8	201.4
B2	Medium RRS	280	360	360	280	203.8
	Medium RPS	280	36	360	41.6	199.9
B3	Medium RRS	640	200	360	212.8	125.6
	Medium RPS	640	20	360	32.8	123.2
C1	Small RRS	640	360	360	360	353
	Large RPS	640	36	360	43.1	50.6
C2	Small RRS	280	360	360	280	283
	Large RPS	280	36	360	39.1	51.2
C3	Small RRS	640	200	360	264	263.8
	Large RPS	640	20	360	27.1	27.1

In each test case, we compare the rate allocation results of the separate swarming with that of the universal swarming. For the separate swarming, we use the subgradient algorithm with a minimum spanning tree solution for the subproblem. This is possible since the sessions are separated from each other and the overlay network for each session contains no Steiner nodes. On the other hand, for the universal swarming, we use the algorithm by Charikar *et. al* with tree level 2, as proposed in [8], for getting an approximate minimum-cost tree solution.

Table I summarizes the simulation results for our test cases.[3] Let u_s, u_i, and d_i be the source upload bandwidth, and each receiver's upload and download bandwidth, respectively. The simulation results show that the subgradient algorithm always achieves the optimal rate allocation in separate swarming.[4] Moreover, with the universal swarming, the RPS can obtain the excessive resource of the RRS at small expense of the RRS. When the small RPS is combined with the large RRS, its session rate improves significantly while the large RRS loses a bit of its session rate. When the

[3] In the simulation, we use $U_s(x_s) = \ln(x_s + e)$ as the utility function, and run the subgradient algorithm for 10000 iterations so that we reach convergence for all the cases. The step size rule and the initial step size used in each profile is slightly different from each other. It is hard to apply the same step size rule for all the profiles and reach convergence within 10000 iterations.

[4] In the separate swarming cases, the optimal rate of each session can be easily computed as $\min\{u_s, \min_{1 \leq i \leq L} d_i, (u_s + \sum_{1 \leq i \leq L} u_i)/L\}$ where L is the number of receivers [19].

session sizes of the RRS and RPS are the same, the resulting session rates tend to be equalized, which is a desirable result. When the large RPS is combined with the small RRS, its session rate still improves slightly with negligible impact on the small RRS; this is also desirable since the RRS should not give up its resource if it is not sufficiently abundant.

Finally, the experimental results have shown that the proposed algorithm converges as expected. The details are omitted for brevity.

VI. CONCLUSION

This paper studies the universal swarming technique for content distribution, which allows multiple sessions to help each other to speed up the overall distribution performance. For the relatively static infrastructure-based content distribution, we can model universal swarming as distribution over multiple multicast trees. That is, the data of each session is distributed by a set of multicast trees rooted at the source and spanning all the receivers. Each multicast tree is in general a Steiner tree containing out-of-session nodes. The question is how to optimally allocate rates to the multicast trees to maximize the sum of all sessions' utilities. A distributed subgradient algorithm is developed. Due to the partial linearity of the problem, there is no standard convergence result for the algorithm and the algorithm does not converge in the normal sense. We prove that the subgradient algorithm converges in the time-average sense. Furthermore, the subgradient algorithm involves an NP-hard subproblem of finding a min-cost Steiner tree. We adopt a column generation method with imperfect min-cost tree scheduling. If the imperfect min-cost tree has bounded performance, then our overall utility optimization algorithm converges to a sub-optimum with bounded performance.

REFERENCES

[1] J. Lee and G. de Veciana, "On application-level load balancing in FastReplica," *Computer Communications*, vol. 30, no. 17, pp. 3218–3231, November 2007.

[2] D. Kostić, A. Rodriguez, J. Albrecht, and A. Vahdat, "Bullet: high bandwidth data dissemination using an overlay mesh," in *Proceedings of 19th ACM Symposium on Operating Systems Principles (SOSP '03)*, October 2003, pp. 282–297.

[3] D. Kostić, R. Braud, C. Killian, E. Vandekieft, J. W. Anderson, A. C. Snoeren, and A. Vahdat, "Maintaining high bandwidth under dynamic network conditions," in *Proceedings of USENIX Annual Technical Conference*, 2005, pp. 14–14.

[4] B.-G. Chun, P. Wu, H. Weatherspoon, and J. Kubiatowicz, "ChunkCast: an anycast service for large content distribution," in *Proceedings of the Internaltional Workshop on Peer-to-Peer Systems (IPTPS)*, February 2006.

[5] BitTorrent Website, http://www.bittorrent.com/.

[6] K. Park and V. S. Pai, "Scale and performance in the CoBlitz large-file distribution service," in *Proceedings of the 3rd USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI)*, San Jose, CA, May 2006, pp. 3–3.

[7] X. Zheng, C. Cho, and Y. Xia, "Optimal peer-to-peer technique for massive content distribution," in *Proceedings of IEEE INFOCOM*, 2008, pp. 151–155.

[8] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, "Approximation algorithms for directed Steiner problems," in *ACM-SIAM Symposium on Discrete Algorithms*, San Francisco, California, 1998, pp. 192–200.

[9] L. Zosin and S. Khuller, "On directed Steiner trees," in *ACM-SIAM Symposium on Discrete Algorithms*, San Francisco, California, 2002, pp. 59–63.

[10] M.-I. Hsieh, E. H.-K. Wu, and M.-F. Tsai, "FasterDSP: a faster approximation algorithm for directed Steiner tree problem," *Journal Of Information Science and Engineering*, vol. 22, pp. 1409–1425, 2006.

[11] S. Chen, O. Gunluk, and B. Yener, "The multicast packing problem," *IEEE/ACM Transaction on Networking*, vol. 8, no. 3, pp. 311– 318, June 2000.

[12] K. Jansen and H. Zhang, "An approximation algorithm for the multicast congestion problem via minimum steiner trees," in *In 3rd International Workshop on Approximation and Randomized Algorithms in Communication Networks ARANCE*, 2002, pp. 152–164.

[13] Y. Wu, P. A. Chou, and K. Jain, "A comparison of network coding and tree packing," in *The Proceedings of IEEE International Symposium on Information Theory (ISIT)*, June 2004, p. 143.

[14] L. Chen, T. Ho, S. H. Low, M. Chiang, and J. C. Doyle, "Optimization based rate control for multicast with network coding," in *Proceedings of IEEE INFOCOM*, 2007, pp. 1163–1171.

[15] D. S. Lun, N. Ratnakar, R. Koetter, M. Mdard, E. Ahmed, and H. Lee, "Achieving minimum-cost multicast: a decentralized approach based on network coding," in *Proceedings of IEEE INFOCOM*, 2005, pp. 1607–1617.

[16] C. A. Oliveira and P. M. Pardalos, "A survey of combinatorial optimization problems in multicast routing," *Computers and Operations Research*, pp. 1953–1981, 2005.

[17] R. Bindal, P. Cao, W. Chan, J. Medval, G. Suwala, T. Bates, and A. Zhang, "Improving traffic locality in BitTorrent via biased neighbor selection," in *Proceedings of the International Conference on Distributed Computing Systems (ICDCS'06)*, 2006, p. 66.

[18] H. Zhang, G. Neglia, D. Towsley, and G. L. Presti, "On unstructured file sharing networks," in *Proceedings of INFOCOM*, May 2007, pp. 2189–2197.

[19] R. Kumar and K. Ross, "Peer-assisted file distribution: the minimum distribution time," in *IEEE Workshop on Hot Topics in Web Systems and Technologies (HOTWEB)*, 2006, pp. 1–11.

[20] P. Key, L. Massouliè, and D. Towsley, "Path selection and multipath congestion control," in *Proceedings of INFOCOM 2007*, May 2007, pp. 143–151.

[21] F. Paganini, "Congestion control with adaptive multipath routing based on optimization," in *The 40th Annual Conference on Information Sciences and Systems*, 2006, pp. 333–338.

[22] X. Lin and N. B. Shroff, "Utility maximization for communication networks with multipath routing," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 766–781, May 2006.

[23] I. Lestas and G. Vinnicombe, "Combined control of routing and flow: a multipath routing approach," in *43rd IEEE Conference on Decision and Control*, 2004, pp. 2390–2395.

[24] X. Zheng, C. Cho, and Y. Xia, "Content distribution by multiple multicast trees and intersession cooperation: optimal algorithms and approximations," in *Manuscript*, 2009, http://www.cise.ufl.edu/∼yx1/publication.html.

[25] D. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 1999.

[26] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, 3rd ed. Wiley-Interscience, 2006.

[27] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer rate control in wireless networks," *IEEE/ACM Transaction on Networking*, vol. 14, no. 2, pp. 302–315, April 2006.

[28] J. Wang, L. Li, S. H. Low, and J. C. Doyle, "Can shortest-path routing and TCP maximize utility," in *Proceedings of INFOCOM*, 2003, pp. 2049–2056.

[29] F. K. Hwang, D. S. Richards, and P. Winter, *The Steiner tree problems*. North-Holland, 1992.

[30] X. Zheng, F. Chen, Y. Xia, and Y. Fang, "A class of cross-layer optimization algorithms for performance and complexity trade-offs in wireless networks," to appear in IEEE Transactions on Parallel and Distributed Systems, http://www.cise.ufl.edu/∼yx1/paper_by_area.html.

[31] P. Bjorklund, P. Varbrand, and D. Yuan, "Resource optimization of spatial TDMA in ad hoc radio networks: a column generation approach," in *Proceedings of INFOCOM*, 2003, pp. 818–824.

[32] M. Johansson and L. Xiao, "Cross-layer optimization of wireless networks using nonlinear column generation," *IEEE Transaction on Wireless Communications*, vol. 5, no. 2, pp. 435–445, Feb. 2006.

[33] S. Kompella, J. E. Wieselthier, and A. Ephremides, "A cross-layer approach to optimal wireless link scheduling with SINR constraints," in *MilCom 2007*, 2007, pp. 1–7.