Optimal Rate Allocation in Overlay Content Distribution

Chuan Wu and Baochun Li

Department of Electrical and Computer Engineering
University of Toronto
{chuanwu, bli}@eecg.toronto.edu

Abstract. This paper addresses the optimal rate allocation problem in overlay content distribution for efficient utilization of limited bandwidths. We systematically present a series of optimal rate allocation strategies by dividing our discussions into four typical scenarios. Based on application-specific requirements, these scenarios reflect the contrast between *elastic* and *streaming* content distribution, with either per-link or per-node capacity constraints. In each scenario, we show that the optimal rate allocation problem can be formulated as a linear optimization problem, which can be solved efficiently in a fully distributed fashion. In simulations, we investigate the convergence of our distributed algorithms in both static and dynamic networks, and demonstrate their efficiency.

Key words: Overlay Network, Rate Allocation, Optimization

1 Introduction

In recent years, content distribution with overlay networks has been proposed to offer more efficient bandwidth usage than that with multiple unicast sessions. To achieve better bandwidth utilization and failure resilience, overlay content distribution over *mesh* topologies has become typical in most recent proposals, which features parallel downloading from multiple overlay nodes.

However, due to the limitation of bandwidth capacities in overlay networks, a critical question remains to be answered in any such content distribution scheme: What is the best way to select upstream peers and allocate flow rates in an overlay topology, such that content can be most efficiently distributed?

To effectively address this question, it is necessary to consider that different content distribution applications may have different optimality goals and constraints for their rate allocation strategies. The content to be distributed can be classified into two broad categories: *elastic* content (*e.g.,* bulk data files), and *streaming* content with specific bit rate requirements (*e.g.,* media streaming). In the case of distributing elastic content, such as file downloading, it is important to optimally select upstream nodes and allocate flow rates so that throughput of content distribution can be maximized. In the case of distributing media streams, the required streaming rate needs to be sustained for all receivers in active sessions. Besides, in both cases, capacity constraints in overlay networks may lie in

the overlay links (*link capacity constraints*), due to limited available bandwidth along the link, or the overlay nodes (*node download/upload capacity constraints*), caused by limited node download/upload capacities.

In this paper, we consider both types of content and both assumptions of capacity constraints, and systematically present a series of *optimal rate allocation* strategies in four content distribution scenarios. We show that in each scenario, the optimal rate allocation problem can be modeled into a linear optimization problem, for which efficient and fully decentralized solutions exist.

The remainder of the paper is as follows. In Sec. 2 and Sec. 3, we motivate the optimization formulations in elastic and streaming content distribution scenarios, respectively, and present efficient distributed solution algorithms. We discuss practical concerns of applying the algorithms in Sec. 4. Simulation results are presented in Sec. 5. We discuss related work and conclude the paper in Sec. 6 and Sec. 7, respectively.

2 *eBurst:* Distribution of Elastic Content

In this paper, we consider content distribution sessions in mesh overlay topologies, consisting of one data *source* S and a set of *receivers* in T. Each receiver is served by one or more *upstream nodes*, and may serve one or more *downstream nodes*. Such a topology can be modeled as a directed graph $G = (N, A)$, where N is the set of overlay nodes and A is the set of overlay links. We have $N = S \cup T$.

To distribute elastic content, it is always desirable to achieve maximum throughput in order to minimize the total time to completion. The problem is: *How do we optimally allocate rates on each overlay link to maximize throughput?* We show that such a problem, referred to as *eBurst*, can be formulated as linear programs.

In order to better characterize the multicast flow of a content distribution session, we resort to the notion of *conceptual unicast flows* [1] in formulating the linear programs. A multicast content distribution flow can be conceptually viewed as consisting of multiple unicast flows from the source to each of the receivers. These conceptual unicast flows co-exist in the overlay without contending for capacities, and the actual delivery rate on an overlay link is the maximum of the rates of all the conceptual flows going through the link. In formulating the linear programs, the utilization of conceptual unicast flows is useful to capture the inherent property of a multicast flow, as the conceptual unicast flows follow the nice property of flow conservation at each intermediate node, while the multicast flows do not.

2.1 eBurst with Link Capacity Constraints

We first consider the assumption that capacity constraints lie in the overlay links, which is the case when the bottleneck is in the core of the overlay, such as transcontinental links. Let u_{ij} be the capacity of overlay link (i, j). R denotes the throughput of the content distribution session, *i.e.*, the aggregate receiving rate at each participating peer. x_{ij} is the delivery rate on link (i, j). Let f^t denote the conceptual unicast flow from source S to a receiver t, $|f^t|$ be its flow rate, and f^t_{ij} be the rate of f^t flowing through link (i, j). The *eBurst* problem with

Link Capacity Constraints (LCC) can be formulated as the linear program in Table 1, referred to as the *eBurst LCC* LP.

Table 1. *eBurst LCC* LP

$$\max R$$

subject to

$$\sum_{j:(i,j)\in A} f_{ij}^t - \sum_{j:(j,i)\in A} f_{ji}^t = b_i^t, \quad \forall i \in N, \forall t \in T, \quad (1)$$

$$f_{ij}^t \geq 0, \forall (i,j) \in A, \forall t \in T, \quad (2)$$

$$f_{ij}^t \leq x_{ij}, \forall (i,j) \in A, \forall t \in T, \quad (3)$$

$$0 \leq x_{ij} \leq u_{ij}, \qquad \forall (i,j) \in A, \quad (4)$$

$$R \geq 0,$$

where

$$b_i^t = \begin{cases} R & \text{if } i = S, \\ -R & \text{if } i = t, \\ 0 & \text{otherwise.} \end{cases}$$

In this LP, (1) and (2) model each conceptual flow f^t as a valid network flow, following flow conservations. (3) represents the relation between conceptual flow rates and the actual delivery rate on each link, which is further constrained by link capacities in (4).

There exists an efficient combinatorial algorithm to solve the *eBurst LCC* LP. By reformulating constraints (2), (3) and (4) as

$$0 \leq f_{ij}^t \leq u_{ij}, f_{ij}^t \leq x_{ij} \leq u_{ij}, \forall (i,j) \in A, \forall t \in T,$$

we notice that this LP can be decomposed into $|T|$ maximum flow problems, each corresponding to one conceptual unicast flow f^t, $\forall t \in T$. Therefore, this LP can be solved by computing maximum flows from the source to each of the receivers, and then delivery rate x_{ij} is decided as the maximum of the rates of all the maximum conceptual flows on (i,j). Since the maximum flow problem can be solved by distributed algorithms, such as push-relabel algorithm [2], we are able to derive an efficient decentralized algorithm for the LP, as given in Table 2.

Table 2. Distributed algorithm for *eBurst LCC* LP

1. Compute maximum flow f^t from S to t, $\forall t \in T$, with distributed push-relabel algorithm.
2. Compute the maximum throughput $R = \min_{t \in T} |f^t|$.
3. Compute optimal rates $x_{ij} = \max_{t \in T} f_{ij}^t$, $\forall (i,j) \in A$.

2.2 eBurst with Node Capacity Constraints

When bandwidth bottlenecks occur at the "last-mile" links to the overlay nodes, it is more appropriate to model capacity constraints at each node rather than each link, with maximum upload and download capacities. For node i, let O_i be

its upload capacity and I_i be its download capacity. The linear program with Node Capacity Constraints (NCC) is formulated in Table 3, referred to as the *eBurst NCC* LP.

Table 3. *eBurst NCC* LP

$$\max R$$

subject to

$$\sum_{j:(i,j)\in A} f_{ij}^t - \sum_{j:(j,i)\in A} f_{ji}^t = b_i^t, \quad \forall i \in N, \forall t \in T, \tag{5}$$

$$f_{ij}^t \geq 0, \forall (i,j) \in A, \forall t \in T, \tag{6}$$

$$f_{ij}^t \leq x_{ij}, \forall (i,j) \in A, \forall t \in T, \tag{7}$$

$$\sum_{j:(i,j)\in A} x_{ij} \leq O_i, \qquad \forall i \in N, \tag{8}$$

$$\sum_{j:(j,i)\in A} x_{ji} \leq I_i, \qquad \forall i \in N, \tag{9}$$

$$R \geq 0, x_{ij} \geq 0, \qquad \forall (i,j) \in A,$$

where

$$b_i^t = \begin{cases} R & \text{if } i = S, \\ -R & \text{if } i = t, \\ 0 & \text{otherwise.} \end{cases}$$

This LP regulates delivery rates on the overlay links using node capacities in (8)(9), rather than link capacities in (4). It is not readily decomposable to known combinatorial optimization problems. To obtain a distributed algorithm, we apply Lagrangian relaxation and design the corresponding subgradient algorithm, which is an efficient LP solution technique and can be naturally implemented in a distributed manner.

We have derived a fully decentralized algorithm by applying Lagrangian relaxation to the dual of the *eBurst NCC* LP. Due to space limit, we only provide the main idea to develop the algorithm. For complete details, interested readers are referred to our technical report [3].

We first note that, if we can decide the set of optimal delivery rates, x_{ij}, $\forall (i,j) \in A$, that satisfy (8)(9), the *eBurst NCC* LP boils down to an *eBurst LCC* problem. In order to obtain the optimal values for primal variables x_{ij}, we investigate the variable-constraint correspondence between an LP and its dual, *i.e.*, each primal variable corresponds to one dual constraint. When Lagrangian relaxation is applied to the dual LP, a primal variable is actually the same as the Lagrangian multiplier associated with its corresponding dual constraint. We further understand that with the Lagrangian relaxation technique, the optimal values for the Lagrangian multipliers can be obtained by the subgradient algorithm. Therefore, to acquire x_{ij}, we solve the dual LP of the *eBurst NCC* LP in Table 3 with Lagrangian relaxation and subgradient algorithm, by relaxing the set of dual constraints corresponding to the primal variables x_{ij}, $\forall (i,j) \in A$.

We also observe that the LP in Table 3 has the underlying structure of maximum flow problems. Therefore, due to the primal-dual relationship between

max-flow and min-cut linear programs, its dual LP has the underlying structure of min-cut problems, which we can utilize when solving the dual LP with subgradient algorithm. The complete distributed algorithm is shown in Table 4.

This distributed algorithm has nice combinatorial interpretations. Starting from some initial feasible delivery rates, the optimal rates are derived iteratively. In each iteration, we increase rates on links in the current minimum cut of the network, *i.e.*, links that are saturated with currently allocated rates, and always guarantee node capacity constraints are satisfied by projecting increased rates onto the feasible simplex P'. After this projection, the bandwidth share for non-saturated links, *i.e.*, links that are not in the minimum cut, is reduced while that for saturated links is increased. This refinement repeats itself until the optimal rate allocation on all the links is achieved.

Table 4. Distributed algorithm for *eBurst NCC* LP

1. Initialize rates $x_{ij}[0], \forall (i,j) \in A$, to non-negative values.

2. Repeat the following iteration until the sequence $\{x[k]\}$ converges to x^*:

(1) With $x_{ij}[k]$ as the upper bound of the delivery rate on link $(i,j), \forall (i,j) \in A$, compute the maximum flow from S to $t, \forall t \in T$, with the distributed push-relabel algorithm;

(2) Update x by

— Compute $x' = x[k] + \theta[k] \sum_{t \in T} z^t[k]$, where $\theta[k] = a/(b+ck), a > 0, b \geq 0, c > 0$, and for all $(i,j) \in A$,

$$z_{ij}^t[k] = \begin{cases} 1 \text{ if edge } (i,j) \text{ is in the min cut of the} \\ \quad \text{minimum of all maximum flows from} \\ \quad S \text{ to } t, \forall t \in T \\ 0 \text{ otherwise.} \end{cases}$$

— Project x' onto the feasible simplex

$$P' = \{x | \sum_{j:(i,j) \in A} x_{ij} \leq O_i, \sum_{j:(j,i) \in A} x_{ji} \leq I_i, \forall i \in N, x_{ij} \geq 0, \forall (i,j) \in A\}$$

by $x_{ij}[k+1] = \min(x'_{ij}, \frac{x'_{ij}}{\sum_{m:(i,m) \in A} x'_{im}} O_i, \frac{x'_{ij}}{\sum_{m:(m,j) \in A} x'_{mj}} I_j), \forall (i,j) \in A.$

→ Optimal delivery rates obtained.

3. With x_{ij}^* as the link capacity on link $(i,j), \forall (i,j) \in A$, compute the maximum flow f^t from S to $t, \forall t \in T$, with the distributed push-relabel algorithm.

4. Compute the maximum throughput $R = \min_{t \in T} |f^t|$.

→ Maximum content distribution throughput obtained.

3 *sBurst:* Distribution of Streaming Content

Real-time content streaming, such as live multimedia or stocks quotes, usually demands a fixed streaming rate, r, to sustain the streaming session. For these applications, instead of maximizing throughput, it is desirable to optimize rate allocations to minimize the total *cost* of streaming, while guaranteeing the streaming rate r. More rigorously, if we use c_{ij} to denote the streaming cost

on an overlay link (i,j), our objective is to minimize $\sum_{(i,j)\in A} c_{ij}x_{ij}$. When c_{ij} represents per-link delay, the optimal rate allocation minimizes total delay of the session. When all c_{ij}'s are 1, the total bandwidth usage is minimized and thus the best bandwidth efficiency is achieved by the optimization. Henceforth, this optimization problem is referred to as *sBurst*.

3.1 sBurst with Link Capacity Constraints

The *sBurst* problem with link capacity constraints, referred to as *sBurst LCC* LP, is formulated in Table 5.

Table 5. *sBurst LCC* LP

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij}$$

subject to

$$\sum_{j:(i,j)\in A} f_{ij}^t - \sum_{j:(j,i)\in A} f_{ji}^t = b_i^t, \quad \forall i \in N, \forall t \in T, \quad (10)$$

$$f_{ij}^t \geq 0, \forall (i,j) \in A, \forall t \in T, (11)$$

$$f_{ij}^t \leq x_{ij}, \forall (i,j) \in A, \forall t \in T, (12)$$

$$0 \leq x_{ij} \leq u_{ij}, \qquad \forall (i,j) \in A, \qquad (13)$$

where

$$b_i^t = \begin{cases} r & \text{if } i = S, \\ -r & \text{if } i = t, \\ 0 & \text{otherwise.} \end{cases}$$

This LP can be solved with Lagrangian relaxation and subgradient algorithm, by relaxing constraint group (12). Associating Lagrangian multipliers μ_{ij} with (12), we obtain the Lagrangian dual of the *sBurst LCC* LP:

$$\max_{\mu \geq 0} L(\mu) \qquad (14)$$

where

$$L(\mu) = \min_{P} \sum_{(i,j)\in A} x_{ij}(c_{ij} - \sum_{t\in T} \mu_{ij}^t) + \sum_{t\in T} \sum_{(i,j)\in A} \mu_{ij}^t f_{ij}^t, \qquad (15)$$

and polytope P is defined by constraints (10)(11)(13).

The Lagrangian subproblem in (15) can be decomposed into $|T|$ shortest path problems:

$$\min \sum_{(i,j)\in A} \mu_{ij}^t f_{ij}^t \qquad (16)$$

subject to

$$\sum_{j:(i,j)\in A} f_{ij}^t - \sum_{j:(j,i)\in A} f_{ji}^t = b_i^t, \forall i \in N,$$
$$f_{ij}^t \geq 0, \qquad \forall (i,j) \in A,$$

for each $t \in T$, and a minimization problem

$$\min \sum_{(i,j)\in A} x_{ij}(c_{ij} - \sum_{t\in T} \mu_{ij}^t) \tag{17}$$

subject to

$$0 \le x_{ij} \le u_{ij}, \forall (i,j) \in A.$$

The shortest path problems in (16) can be efficiently solved in a distributed manner by the distributed Bellman-Ford algorithm [4]. The optimal solution to the minimization problem in (17) can be computed as follows:

$$x_{ij} = \begin{cases} 0 & \text{if } \sum_{t\in T} \mu_{ij}^t \le c_{ij}, \\ u_{ij} & \text{if } \sum_{t\in T} \mu_{ij}^t > c_{ij}. \end{cases} \tag{18}$$

In each iteration of the subgradient algorithm, we solve the subproblems in (16) and (17) with the current Lagrangian multiplier values $\mu[k]$. Then we update the Lagrangian multipliers by

$$\mu_{ij}^t[k+1] = \max(0, \mu_{ij}^t[k] + \theta[k](f_{ij}^t[k] - x_{ij}[k])), \forall (i,j) \in A, \forall t \in T,$$

where θ is a prescribed sequence of step sizes satisfying:

$$\theta[k] > 0, \lim_{k\to\infty}\theta[k] = 0, \text{ and } \sum_{k=1}^{\infty} \theta[k] = \infty.$$

Since the primal values in the optimal solution of the Lagrangian dual are not necessarily optimal to the primal LP, we further apply the algorithm introduced by Sherali *et al.* [5] to recover the optimal primal values. At the k^{th} iteration, we compose a primal iterate $\widetilde{f_{ij}^t}[k]$ via

$$\widetilde{f_{ij}^t}[k] = \sum_{h=1}^{k} \lambda_h^k f_{ij}^t[h], \tag{19}$$

where $\sum_{h=1}^{k} \lambda_h^k = 1$ and $\lambda_h^k \ge 0$, for $h = 1, \ldots, k$.

Table 6. Distributed algorithm on node j for *sBurst LCC* LP

1. Initialize Lagrangian multipliers $\mu_{ij}^t[0]$, $\forall i : (i,j) \in A, \forall t \in T$, to non-negative values.

2. Repeat the following iteration until sequence $\{\mu[k]\}$ converges to μ^*, $\{\widetilde{f}[k]\}$ converges to $\widetilde{f}^*$: $\forall i : (i,j) \in A, \forall t \in T$

(1) Compute $f_{ij}^t[k]$ by distributed Bellman-Ford algorithm;

(2) Compute $x_{ij}[k]$ by Eqn. (18);

(3) Compute $\widetilde{f_{ij}^t}[k] = \sum_{h=1}^{k} \frac{1}{k} f_{ij}^t[h] = \frac{k-1}{k}\widetilde{f_{ij}^t}[k-1] + \frac{1}{k} f_{ij}^t[k]$;

(4) Update Lagrangian multiplier $\mu_{ij}^t[k+1] = \max(0, \mu_{ij}^t[k] + \theta[k](f_{ij}^t[k] - x_{ij}[k]))$;

3. Compute optimal rate $x_{ij}^* = \max_{t\in T} \widetilde{f_{ij}^t}^*$, $\forall i : (i,j) \in A$.

In our algorithm, we choose the step length sequence $\theta[k] = a/(b+ck), \forall k, a > 0, b \geq 0, c > 0$, and convex combination weights $\lambda_h^k = 1/k, \forall h = 1, \ldots, k, \forall k$. These guarantee the convergence of our subgradient algorithm; they also guarantee that any accumulation point $\widetilde{f}^*$ of the sequence $\{\widetilde{f}[k]\}$ generated via (19) is an optimal solution to the primal problem in Table 5 [5].

Now we can design our distributed algorithm to solve *sBurst LCC* LP. We delegate the computation tasks on link (i, j) to be carried out by incident node j. The algorithm to be executed by each node is given in Table 6.

3.2 sBurst with Node Capacity Constraints

The *sBurst* problem with node capacity constraints, referred to as *sBurst NCC* LP, is formulated in Table 7.

Table 7. *sBurst NCC* LP

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}$$

subject to

$$\sum_{j:(i,j)\in A} f_{ij}^t - \sum_{j:(j,i)\in A} f_{ji}^t = b_i^t, \quad \forall i \in N, \forall t \in T,$$

$$f_{ij}^t \geq 0, \forall (i,j) \in A, \forall t \in T,$$

$$f_{ij}^t \leq x_{ij}, \forall (i,j) \in A, \forall t \in T, \quad (20)$$

$$\sum_{j:(i,j)\in A} x_{ij} \leq O_i, \qquad \forall i \in N,$$

$$\sum_{j:(j,i)\in A} x_{ji} \leq I_i, \qquad \forall i \in N,$$

$$x_{ij} \geq 0, \qquad \forall (i,j) \in A,$$

where

$$b_i^t = \begin{cases} r & \text{if } i = S, \\ -r & \text{if } i = t, \\ 0 & \text{otherwise.} \end{cases}$$

This LP can be solved with similar Lagrangian relaxation techniques as solving the *sBurst LCC* LP, by relaxing (20). The only difference is that the resulting minimization subproblem is defined differently:

$$\min \sum_{(i,j)\in A} x_{ij}(c_{ij} - \sum_{t\in T} \mu_{ij}^t) \quad (21)$$

subject to

$$\sum_{j:(i,j)\in A} x_{ij} \leq O_i, \forall i \in N,$$

$$\sum_{j:(j,i)\in A} x_{ji} \leq I_i, \ \forall i \in N,$$

$$x_{ij} \geq 0, \qquad \forall (i,j) \in A,$$

which is an inequality constrained transportation problem, and can be solved by distributed auction algorithm [6]. Thus, we can also design a distributed algorithm for *sBurst NCC* LP, as summarized in Table 8.

Table 8. Distributed algorithm on node j for *sBurst NCC* LP

1. Initialize Lagrangian multipliers $\mu_{ij}^t[0]$, $\forall i : (i,j) \in A, \forall t \in T$, to non-negative values.

2. Repeat the following iteration until sequence $\{\mu[k]\}$ converges to μ^*, $\{\widetilde{f}[k]\}$ converges to $\widetilde{f}^*$: $\forall i : (i,j) \in A, \forall t \in T$

(1) Compute $f_{ij}^t[k]$ by distributed Bellman-Ford algorithm;

(2) Compute $x_{ij}[k]$ by distributed auction algorithm;

(3) Compute $\widetilde{f_{ij}^t}[k] = \frac{k-1}{k}\widetilde{f_{ij}^t}[k-1] + \frac{1}{k}f_{ij}^t[k]$;

(4) Update Lagrangian multiplier $\mu_{ij}^t[k+1] = \max(0, \mu_{ij}^t[k] + \theta[k](f_{ij}^t[k] - x_{ij}[k]))$;

3. Compute optimal rate $x_{ij}^* = \max_{t \in T} \widetilde{f_{ij}^t}^*$, $\forall i : (i,j) \in A$.

4 Algorithm Execution in Dynamic Overlays

In an overlay session characterized by dynamics, the proposed distributed algorithms are also invoked in a dynamic manner. When a node *joins a session*, it is bootstrapped with a set of upstream nodes. It then starts downloading with the available upload capacities acquired from them. Meanwhile, it requests the source to recompute the optimal rate allocation. When a node *departs* from a session or *fails*, an affected downstream node attempts to acquire additional bandwidths from its remaining upstream nodes. Meanwhile, it requests the source to recompute the optimal rate allocation.

At the source, when it receives more than a certain number of requests for recomputation, it broadcasts such a request to all the nodes, which activate a new round of execution of the distributed algorithm, while continuing to download at the original rates. Note that in such a dynamic environment, the execution of a distributed algorithm always starts from the previous optimal values (rather than from the very beginning when all values are initialized to any non-negative values, such as zeros), thus expediting its convergence. After the distributed algorithm converges, all the nodes adjust their download rates to the new optimal values.

5 Performance Evaluation

We next conduct an empirical study of the distributed optimization algorithms. All simulations are conducted over random network topologies generated with BRITE [7] topology generator based on power-law degree distributions. The average number of neighbors per node in the topologies is six. For link-constrained problems, link capacities are generated with heavy-tailed distributions between 100 Kbps and 4 Mbps; for node-constrained problems, each node has $1.5 - 4.5$ Mbps of download capacity and $0.6 - 0.9$ Mbps of upload capacity. For *sBurst* problems, streaming of a 300 Kbps bitstream is simulated and cost coefficients are random numbers chosen from $(0, 3)$.

5.1 Convergence in Static Networks

To investigate the scalability of our optimal rate allocation algorithms, we first evaluate their convergence speed in static networks of different sizes. For *eBurst*

LCC LP, we have shown in Table 2 a purely combinatorial algorithm, which can derive the solution efficiently. Here, we are more concerned with the efficiency of the iterative subgradient algorithms to solve the other three problems, given in Table 4, 6, 8, respectively.

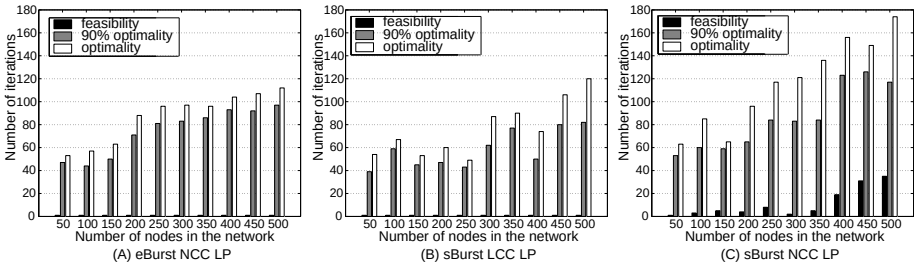

Fig. 1. Convergence speed in static networks for distributed algorithms in Table 4, 6, 8.

Fig. 1 shows that for all three problems, with the increase of network sizes, the numbers of iterations their algorithms take to achieve optimality only increase slowly, thus not affecting algorithm scalability. In all cases, the algorithms converge to feasibility within only a few rounds. Even the convergence to 90% optimality is much faster, within 20% fewer rounds than those required for convergence to optimality. Therefore, in realistic networks, we can obtain a feasible solution to a certain degree of optimality in a much shorter time, when it is not necessary to achieve absolute optimality.

5.2 Convergence in Dynamic Networks

We next investigate the algorithm convergence in practical dynamic environments. Due to space limit, we only show the results obtained by the *eBurst NCC* algorithm in Table 4, while other algorithms produce similar results.

In this experiment, 200 nodes sequentially join an elastic content distribution session, and then start to depart when their downloads are completed. The distributed algorithm is invoked every 10 node joins or departures. As discussed in Sec. 4, the algorithm always runs from the previously converged optimal rates when it is invoked.

We show the number of additional iterations required to converge to new optimal values from the previous ones in node joining and departure phases in Fig. 2 (A) and (B), respectively. We find that, compared to running from the very beginning in the cases of static networks of the same sizes, our dynamic execution of the algorithm converges much faster. Independent of the current network sizes, it always takes less than 15 iterations to converge, in both joining and departure cases. While dynamic networks are more akin to realistic scenarios, this suggests our optimal rate allocation algorithms can deliver good performance and provide excellent scalability in practice.

In addition, we illustrate maximum throughput of the dynamic session, R, in Fig. 3. In Fig. 3 (A), at the beginning of the node joining phase, the throughput

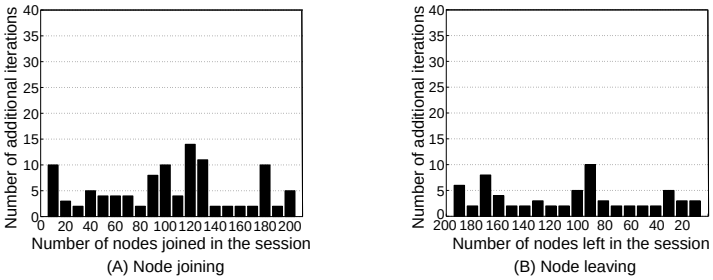

Fig. 2. Convergence speed in a dynamic network for *eBurst NCC* algorithm in Table 4.

drops because of the competition of more nodes for the available upload capacities in the network. Later, when more nodes have joined, more upload capacities are provided to the session, and thus the throughput gradually increases. During the node departure phase in Fig. 3 (B), due to similar reasons, the throughput first shows a decreasing trend, and then rises when only a few nodes are left.

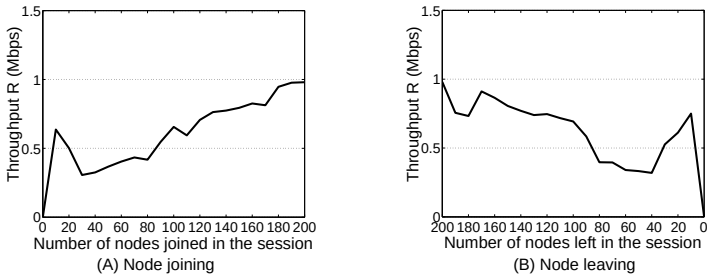

Fig. 3. Throughput achieved in a dynamic network with *eBurst NCC* algorithm in Table 4.

6 Related Work

On the topic of overlay content distribution, mesh-based approaches have become typical in most recent proposals [8–10]. Disseminating large-scale content on mesh topologies, their parallel transfers make them possible to deliver at fundamentally higher bandwidth and reliability, *without* the cost of constructing multicast trees.

With respect to rate allocation in mesh overlay topologies, most existing work either relies on TCP, or employs various heuristics without formulating the problem theoretically. Compared to our optimal rate allocation, their rate allocation falls short of achieving global optimality. There is no way to guarantee that maximum throughput is achieved at all nodes, or a required streaming rate is provided to all at the lowest cost.

There are a few exceptions that formulate the problem into optimization models and propose distributed solutions [1, 11, 12]. Our work is original in that we systematically model all the typical content distribution scenarios, and design

efficient algorithms to solve the formulated optimization problem combinatorially or numerically, in a fully distributed manner. In addition, we discuss execution of the algorithms in practical dynamic environments. This has not been addressed in previous optimization-based approaches, most of which are largely theoretical in nature.

7 Conclusion

The problem of interest in this paper is to design efficient distributed algorithms for optimal rate allocation under all typical scenarios of overlay content distribution. For this purpose, we formulate rate allocation problems into linear programs, which optimize bandwidth utilization towards a variety of objectives, and develop fully decentralized algorithms to efficiently compute the optimal link rates. We believe such an optimal rate allocation algorithm is critical to any schemes of overlay content distribution. As ongoing work, we are investigating the combination of optimal rate allocation with efficient distribution schemes, and its application in realistic networks.

References

1. Li, Z., Li, B.: Efficient and Distributed Computation of Maximum Multicast Rates. In: Proc. of IEEE INFOCOM 2005. (March 2005)
2. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice Hall (1993)
3. Wu, C., Li, B.: Optimal Rate Allocation in Overlay Content Distribution. Technical report, http://iqua.ece.toronto.edu/papers/ratealloc.pdf (Oct 2006)
4. Bertsekas, D.P., Gallager, R.: Data Networks, 2nd Ed. Prentice Hall (1992)
5. Sherali, H.D., Choi, G.: Recovery of Primal Solutions when Using Subgradient Optimization Methods to Solve Lagrangian Duals of Linear Programs. Operations Research Letter **19** (1996) 105–113
6. Bertsekas, D.P., Castanon, D.A.: The Auction Algorithm for the Transportation Problem. Annals of Operations Research **20** (1989) 67–96
7. Medina, A., Lakhina, A., Matta, I., Byers, J.: BRITE: Boston University Representative Internet Topology Generator. Technical report, http://www.cs.bu.edu/brite (2000)
8. Kostic, D., Rodriguez, A., Albrecht, J., Vahdat, A.: Bullet: High Bandwidth Data Dissemination Using an Overlay Mesh. In: Proc. of the 19th ACM Symposium on Operating Systems Principles (SOSP) 2003. (October 2003)
9. Sherwood, R., Braud, R., Bhattacharjee, B.: Slurpie: A Cooperative Bulk Data Transfer Protocol. In: Proc. of IEEE INFOCOM 2004. (March 2004)
10. Zhang, X., Liu, J., Li, B., Yum, T.P.: CoolStreaming/DONet: A Data-Driven Overlay Network for Live Media Streaming. In: Proc. of IEEE INFOCOM 2005. (March 2005)
11. Lun, D.S., Ratnakar, N., Koetter, R., Medard, M., Ahmed, E., Lee, H.: Achieving Minimum-Cost Multicast: A Decentralized Approach Based on Network Coding. In: Proc. of IEEE INFOCOM 2005. (March 2005)
12. Adler, M., Kumar, R., Ross, K.W., Rubenstein, D., Suel, T., Yao, D.D.: Optimal Peer Selection for P2P Downloading and Streaming. In: Proc. of IEEE INFOCOM 2005. (March 2005)