Min-Cost Live Webcast under Joint Pricing of Data, Congestion and Virtualized Servers

Rui Zhu, Di Niu
Department of Electrical and Computer Engineering
University of Alberta

Baochun Li
Department of Electrical and Computer Engineering
University of Toronto

Abstract*—Live video webcast streams the same live content from its source to a large number of Internet clients through CDN networks, and may consume large volumes of traffic and contribute significantly to Internet congestion during popular events such as the FIFA World Cup. A fair pricing scheme to charge webcast operators should take into account the total amount of data transferred in the network, the cost of running virtualized server instances on the selected CDN edge locations, as well as the potential congestion level the webcast may impose on the underlying Internet. In this paper, we propose a comprehensive pricing model for video webcast operators based on all of the above factors. In particular, we model the congestion level of the webcast as the sum of bandwidth-delay products on all links in the formed video distribution overlay, which indicates the total number of "waiting packets" occupying the Internet. Under such a pricing model, we formulate the min-cost webcasting via at least k CDN servers as a problem we call "k-Node-Weighted Steiner Tree", for which we give the first polynomial-time approximation scheme with an approximation ratio of $O(\ln n)$, where n is the number of participating clients. By relaxing the pricing and cost model, we point out the relationship of the special cases of problem with a number of famous problems including Uncapacitated Facility Location problem(UFL) and Node-Weighted Steiner Tree problem(NWST). We verify the effectiveness of our algorithm at reducing the cost of live webcast based on real delay traces collected from PlanetLab and the Seattle project with pricing parameters synthesized from Amazon Web Service.

I. Introduction

The rapid growth in residential broadband capacity has enabled delivery of high-quality media content over the Internet at a large scale. As a typical application, live video webcast streams a popular event, such as a lecture, a festival celebration or sport game, captured from its source to Internet clients across a wide region or even the globe. Instead of directly of delivering content from the source, live webcast relies on content delivery networks (CDNs) such as Akamai and Amazon CloudFront [1], where content is transferred to selected CDN edge locations on a high-speed backbone network, which then deliver the content to their local clients for viewing. Such a CDN-based streaming architecture, while effectively reducing service latency, may transfer large volumes of data, contribute significantly to Internet congestion, and engage many CDN server resources, especially during popular events such as the FIFA World Cup, making the cost and pricing model of live webcast complicated to analyze.

Most current data pricing policies charge end users for the consumed traffic, including live video traffic, according to a monthly flat rate plan, a usage fee, or a combination of both. Such policies, however, has placed excessive burden on clients and decrease their willingness to view multimedia content. This fact has motivated some mobile and Internet Service Providers (ISPs) to bundle content into their data plans, in a recently initiated trend called *content sponsoring*. For instance, a major ISP in Canada, Telus, provided a free six-month subscription to *Rdio* (music streaming) to its subscribers, who purchased a *Rdio*-supported smartphone and data plan [2]. Such bundled plans are possible because many content/application providers are willing to subsidize customers by directly paying ISPs for bandwidth and other costs. When customers use these applications, their traffic usage may not count towards the data caps in their subscribed plan or may be charged differently, encouraging customers to use the bundled multimedia applications more often.

This paper studies the cost and pricing model for live webcast operators, as business clients of ISPs, CDNs, and even cloud providers, and ask the question—how should webcast operators pay for the video delivery service supported on the wide-area infrastructure formed by these different infrastructure providers, considering the cost they have incurred on it? We believe that a fair pricing scheme to charge webcast operators should take into account the total amount of data transferred in the entire network, the cost of using CDN servers for traffic relaying (e.g., by running VM-based or container-based server instances on selected CDN edge nodes), as well as the potential congestion level the webcast may impose on the underlying Internet.

Besides usage-based pricing, in our joint pricing model, we have placed additional emphasis on the congestion-level that the webcast session may incur on the Internet. The reason is that if "tariff" on congested Internet routes can be effectively set, the webcast operators will be encouraged to select less congested routes when constructing their session-specific media streaming topology. This is analogous to charging each vehicle a higher toll on congested routes in a road network. Such a congestion-aware pricing, if implemented, can help estimate and reduce the negative network externalities that a webcast session imposes on other applications, therefore alleviating congestion and improving the overall quality of service provided by the ISPs to all business and personal clients in general.

In particular, we require a webcast operator to pay a congestion fee per overlay link in proportion to the *bandwidth-delay* product on the link. In a toy example, given two overlay routes from node A to node B, under the same throughput, the application will choose the route with a shorter delay, thus minimizing the "waiting" packets on the overlay link, which is on expectation equal to the bandwidth-delay product according to queueing theory. This is comparable to congestion control on a road network by imposing a toll on each road according to the number of vehicles occupying the road. Under such congestion pricing, each webcast provider will choose servers and relay nodes from the large pool of content distribution network (CDN) nodes and datacenters to form their min-cost media streaming topologies, collectively minimizing the "waiting packets" occupying the Internet.

The question is—how should each webcast provider compute its min-cost overlay network topology? Under joint pricing of data, congestion and server cost, we formulate the selection of at least k CDN servers and min-cost topology formation as a problem that we call "k-Node-Weighted Steiner Tree"(k-NWST). Although the special cases of this problem (by ignoring a part of the entire cost model or relaxing the server number constraint) turn out to be the well-known Uncapacitated Facility Location (UFL) problem, the Steiner Tree (ST) problem, and the Node-Weighted Steiner Tree (NWST) problem, yet there is no existing known solution to the general k-NWST.

In this paper, we give the first polynomial-time approximation scheme (PTAS) to this problem with an approximation ratio of $2.7 \ln n + 1$, where n is the number of participating clients in the live webcast. We verify the effectiveness of our algorithm in reducing the cost of live webcast based on real-world delay traces collected from PlanetLab and the Seattle project [3], which include both server nodes and personal devices, with pricing parameters synthesized from Amazon Web Service [1].

The remainder of this paper is organized as follows. In Sec. II, we propose our joint pricing scheme for data, congestion and virtualized servers. To model and study the response of live webcast operators to the proposed pricing policy, we formulate the k-NWST problem in Sec. III and discuss the hardness of it in Sec. IV by ignoring part of constraints or objectives. We propose a polynomial-time approximation scheme (PTAS) in Sec. V for an application to form its min-cost multicast topology. Simulation results are presented in Sec. VI. Finally, we present related work in Sec. VII and conclude the paper in Sec. VIII.

II. A Joint Pricing Model for Data, Congestion and Servers

In this section, we introduce our joint pricing model for data, congestion and virtualized servers. The pricing model is important since it will affect the each webcast operator's choice of CDN edge nodes, the assignment of all the clients to these nodes and the entire content distribution topology formed from the source to clients. Before motivating the necessity of congestion pricing for content/application providers, we first review existing data pricing policies mostly targeted for end users. We also consider the cost of running virtualized server instances on selected CDN nodes.

A. Conventional Usage-based Data Pricing

Current Internet or mobile service users are mostly charged in the so-called usage-based pricing model [4], [5]: users are charged either 1) a fixed monthly fee under a "Capped" plan, or 2) a "Metered" fee which is proportional to the volume of data usage, or 3) a combination of 1) and 2) in a "Cap then metered" plan, in which a user pays a flat fee up to a certain cap on data usage, beyond which the user is charged in proportion to the usage. For example, in 2010, AT&T introduced a $15/month data plan for 200 MB and $25/month for 2 GB, with different rates of overage charges for the two tiers [6], [7]. However, it is a widespread concern that the usage-based pricing "charges customers irrespective of congestion levels in the network, and still fails to overcome the problem of large peak load costs incurred from many users crowding on the network at the same time. [5]"

B. Congestion Pricing Inspired by Road Pricing

Our pricing policy for webcast operators is not only based on the total number of data transferred in the formed media distribution network, but it also includes a "tariff" component based on the degree of congestion that the application incurs on the Internet. However, there are several challenges to designing a congestion-aware pricing policy:

- It is hard to measure the congestion level of each link: simply measuring bandwidth or delay is insufficient.
- The webcast session may use a large number of underlying physical links on the Internet; it is impossible to monitor all the link states.
- Congestion levels change over time and are hard to be monitored dynamically, if not impossible.

Our proposed congestion-aware pricing policy is inspired by congestion-specific road pricing [5]. Transportation networks are among the first networks that adopt some form of congestion pricing, e.g., in Hong Kong [8]. One of the most natural road pricing policies is *Distance traveled pricing*, in which a vehicle pays for the distance it has traveled. *Congestion-specific pricing* [5], a dynamic pricing policy considered for Cambridge, UK, combines the distance traveled and the time spent to travel that distance; it makes the price rate per mile dependent on the speed at which the vehicle travels. As a result, each vehicle pays a fee that is proportional to *a function of both the distance traveled and its speed*.

Given a set of end-user clients, a webcast operator can connect the users, CDN nodes and the source into an overlay network, while optimally selecting a limited number of CDN nodes as content relay servers. Our congestion pricing policy will charge the webcast operator a *per-minute price rate* on each link in its constructed overlay network. Such a per-minute rate is proportional to the *product of the webcast session's throughput on that overlay link and its packet transmission* *delay on the link*. As a result, the webcast operator needs to pay a per-minute congestion fee that is proportional to the *sum of bandwidth-delay products on all the overlay links* in the media streaming topology it has constructed.

The proposed congestion pricing policy has several advantages: *First,* the bandwidth-delay product of a certain application on a link indicates the congestion on it, while either throughput or delay alone does not. In fact, by Little's law in queuing theory, the bandwidth-delay product is the amount of data occupying the link at any given time, i.e., the "waiting data" that has been transmitted but not yet acknowledged. Congestion will occur, if the bandwidth-delay product of the application on the link approaches the inherent bandwidth-delay product that this link can accommodate. *Second,* the proposed congestion pricing is oblivious to specific underlying physical links, routers and bridges through which an application's packets travel. By only considering the end-to-end throughput and delay on each of the overlay links among users and servers, the policy abstracts away from the detailed measurements on physical links. *Third,* it is relatively easy to record both the throughput and delay of a webcast session on each overlay link by just introducing software metering functions on CDN nodes.

Once the congestion pricing policy is posted, every webcast operator will have the economic incentive to minimize the sum of bandwidth-delay products on all its overlay links, or in other words, minimizing its "waiting data" occupying the Internet.

C. Cost of Virtualized Servers

In the media distribution network formed by the webcast operator, the cost of running software functions on the selected CDN nodes must be considered. Such software is responsible for downloading content from the source, recoding the content to adapt to a bit-rate that match each client's network capacity, and transferring the encoded media to clients. Since each CDN edge node is a small datacenter, the cost of running load-balancers among servers collocated at the same CDN edge node must also be considered. Due to VM-based or lightweight container-based virtualization technologies, the content and processing functions of each webcast operator can be encapsulated into specific VMs and containers. Virtualized server instances at each CDN node not only ensures isolation among different applications and webcast operators, but also enable easy monitoring the resource or energy usage associated with each webcast provider. As a detailed server cost model is not the focus of this paper, for simplicity of analysis, we assume there is a fixed per-minute server cost at each CDN node employed by the given webcast provider.

III. MIN-COST MULTICAST AS k-NODE-WEIGHTED STEINER TREE

In this section, we formulate the min-cost webcast (multicast) problem as an integer program with at least k CDN servers employed. The webcast operator will not serve content from the source to end-users directly, but do so by duplicating the content on at least k relay CDN servers, henceforth also referred to as *relay servers*. This is to take advantage of the powerful CDN infrastructure to reduce delivery latency and augment bandwidth capacity. In particular, the general problem is to select these relay servers, find the topology connecting them and the source (allowing communication between relay servers), and then deciding which one of relay servers should serve each client, in order to minimize a total cost depending on the congestion cost, data cost and server cost.

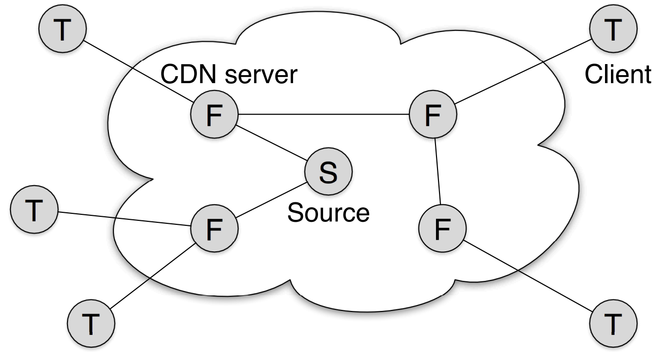

Fig. 1. The system model for live webcast. A source S streams media to clients using the selected CDN edge nodes as relay servers, where relay servers can communicate with each other to reduce the media delivery cost.

Let S denote the source node who wishes to multicast the same media stream to n clients $T_1, \ldots, T_n$, with the client set being $T = \{T_1, \ldots, T_n\}$. We denote $F = \{F_1, \ldots, F_m\}$ the pool of available CDN nodes, from which at least k CDN nodes should be chosen as relay servers to forward the received media stream. We use a complete graph $G = (V, E)$ to denote all CDN nodes, where $V = F \cup \{S\}$ denotes the set of source node and relay CDN servers, and E denotes the set of undirected edges inter-connecting them.

For a server-client pair (i, j), where $i \in F$ and $j \in T$, we add an edge (i, j) between them and the graph can be extended as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = V \cup T$ and $\mathcal{E} = E \cup \{(i, j) : i \in F, j \in T\}$. For all $e \in \mathcal{E}$, we assign a positive real-valued function $c : \mathcal{E} \to \mathbb{R}^+$ to denote the link cost which is proportional to the link delay. This delay cost will be used to calculate the congestion fee, which is proportional to the throughput-delay product on the link. When transferring replicated streams, each link will have the same throughput as the source video bit rate r. Hence, the congestion fee, which is the sum of bandwidth-delay products on all selected links, is simply proportional to the sum of delays on all the *selected* links $e \in \mathcal{E}$ in the network.

Each client $j \in T$ has to connect to exactly one relay server $i \in F$ to receive media. When a client j is served by a server i, we say that j *belongs* to cluster i, or simply belongs to i. We use an integer variable $x_{i,j}$ to indicate this belonging relationship, and let $x_{i,j} = 1$ if j belongs to i, and let $x_{i,j} = 0$ otherwise. Since each client j can only belong to one (relay) server, we have a constraint $\sum_{i \in F} x_{i,j} = 1$ for all clients $j \in T$.

For each server $i \in F$, the webcast operator should pay a per-minute server fee f_i, where f_i can be deemed as the *opening cost* for server i. We let $y_i = 1$ if server i is employed and $y_i = 0$ otherwise. It is clear that we have $x_{i,j} \leq y_i$, since $x_{i,j}$ can be positive only when server i is used.

We allow content transmission between relay servers that might further reduce the total cost in the network. We use $z_{i,i'}$ to denote the connectivity between a server i (including the source) and server i', where $i, i' \in V$. Then, $z_{i,i'} = 1$ indicates that i will transmit the replicated media stream to i', i.e., the inter-server edge (i, i') is selected in the final distribution network.

Our min-cost multicast problem can then be formulated as the following program that we call the "k-Node-Weighted Steiner Tree":

$$\begin{array}{lrcll} \underset{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}}{\text{minimize}} & \multicolumn{4}{l}{\sum_{e\in E} c_e z_e + \sum_{i\in F} f_i y_i + \sum_{i\in F, j\in T} c_{i,j} x_{i,j}} \\ \text{subject to} & \sum_{e\in\delta(N)} z_e & \geq & y_i, & (\forall i \in N \subseteq F) \\ & \sum_{i\in F} x_{i,j} & = & 1, & (\forall j \in T) \\ & x_{i,j} & \leq & y_i, & (\forall i \in F, j \in T) \\ & \sum_{i\in F} y_i & \geq & k, & \\ & x_{i,j} & \in & \{0,1\}, & (\forall i \in F, j \in T) \\ & y_i & \in & \{0,1\}, & (\forall i \in F) \\ & z_e & \in & \{0,1\}. & (\forall e \in E) \end{array} \tag{1}$$

The objective function is the sum of a congestion fee $\sum_{e\in E} c_e z_e + \sum_{i\in F, j\in T} c_{i,j} x_{i,j}$ and the server opening cost $\sum_{i\in F} f_i y_i$ (including data cost to be explained soon). Recall that when transferring replicated streams, each link will have the same throughput as the source video bit rate r. Hence, the congestion fee, which is the sum of bandwidth-delay products on all links $\mathcal{E}$, is simply proportional to, the sum of delays on all chosen links $e \in \mathcal{E}$ in the network, which has two parts: the congestion fee in the inter-server network $\sum_{e\in E} c_e z_e$ and the congestion fee in the client-to-server network $\sum_{i\in F, j\in T} c_{i,j} x_{i,j}$.

The total amount of data transferred in the multicast network is implicitly incorporated in the server opening cost $\sum_{i\in F} f_i y_i$. It turns out that the optimal solution forms a tree, for reasons to be clear soon. If this is the case, the total data transferred per unit time in the network is the total number of selected edges times the video bit rate r, which is $\sum_{i\in F} y_i r + nr$. Since the total data transferred per unit time for the clients is a constant nr, given the number of clients n. The total data cost only depends on $\sum_{i\in F} y_i r$, and thus can be incorporated into the server opening cost $\sum_{i\in F} f_i y_i$ without changing the problem structure.

We require the number of employed relay servers to be at least k, since in reality, the webcast operator usually needs to rely on at least a certain number of CDN nodes, which are geographically distributed in various areas to reduce content delivery delay as well as to increase robustness to failures. We do not consider an upper bound on the number of employed relay servers, since server opening cost is already considered in the minimization objective of problem (1), which will limit the number of servers actually used. In fact, in most cases, the optimal solution of (1) is achieved when exactly k servers are used.

The first constraint in problem (1) says that the optimal solution for such low-cost multicast must be a tree. We now explain the rationale behind it. Recall that the objective is to multicast media to all clients, while the relay servers store-and-forward all received packets to downlink nodes, including other server nodes and clients. If there is a path between any two nodes, adding another link between them would definitely increase the total cost. Therefore, the chosen relay servers must be interconnected as a tree rooted at S, and each cut $CUT(N, \bar{N})$ should include some edges in the tree, where $N \subseteq V \backslash \{S\}$ is a subset of all server nodes excluding the source node S. More formally, if such a cut is denoted by $\delta(N)$, we must have $z_e = 1$ for at least one edge $e \in \delta(N)$, and we thus have

$$\sum_{e\in\delta(N)} z_e \geq y_i, \ \forall i \in N \subseteq V \backslash \{S\}.$$

In our problem, the set of relay servers F equals to $V \backslash \{S\}$. Clearly, the selected edges in the extended graph $\mathcal{G}$ including clients is still a tree, since clients are all leaves, each connected to only one relay server.

IV. Hardness of the Problem and Special Cases

The k-Node-Weighted Steiner Tree problem (1) formulated in the previous section can be proved to be NP-hard, because several special cases of this problem by ignoring part of the cost or some constraint are well-known NP-hard problems. In other words, the k-Node-Weighted Steiner Tree problem (1) is a general extension of some famous NP-hard problems. In this section, we will discuss how to reduce our problem into the well-known UFL problem, ST problem, NWST problem.

A. Steiner Tree Problem: No Server/Data Cost

If there is no server opening cost and no data cost (as has been mentioned, the data cost can be deemed as a part of the server opening cost), then $f_i = 0$ for all relay server $i \in F$. In this case, the task to minimize the congestion cost is equivalent to finding a min-cost tree in the extended graph $\mathcal{G}$ where the source and all clients are required to be included. This is indeed the Steiner Tree problem which has been widely investigated in many fields and plays an important role in the study of theoretical computer science. The Steiner Tree problem is proved to be NP-hard, but can be constantly approximated in polynomial time. The best approximation ratio so far is 1.386 [9], while it is proved that approximating within a factor of 1.0105 is NP-hard [10].

B. Uncapacitated Facility Location Problem: No Inter-Server Connection

The Facility Location problem is to open some facilities in the set of available facilities and assign clients to them in order to minimize the total cost. This problem is widely applied in many economical decision problems, where facilities can be manufacturing plants, depots, warehouses, hospitals, etc. In the field of networking, servers can be deemed as facilities, and the clients in similar definition. The term of "uncapacitated" means that there is no limitation on how many clients each facility can serve.

We show that some special cases of our problem can be treated as the UFL problem. The first case is when the delay among relay servers can be neglected and the first term in problem (1) can be eliminated. Another case is when relay servers are not permitted to transfer to each other. In this case, edges in E only consist of outgoing edges from the source node and the first term of the objective function in problem (1) now becomes:

$$\sum_{e\in E} c_e z_e = \sum_{i\in F} c_{s,i} z_{s,i} = \sum_{i\in F} c_{s,i} y_i.$$

Thus, we can redefine the relay server opening cost by summing $c_{s,i}$ and f_i for all selected relay servers i.

The UFL problem is proved to be NP-hard by reduction from the set cover problem. Without assumptions on the latencies between servers and clients, the problem can be approximated to within a factor $O(\log n)$. If the latencies satisfy the triangle inequality, i.e., $c_{i,j} + c_{i',j} + c_{i',j'} \geq c_{i,j'}$ for all $i, i' \in F$, and for all $j, j' \in T$, the currently best known PTAS can achieve an approximation ratio of 1.488 [11].

C. Node-Weighted Steiner Tree Problem: No Server Number Constraint

If we assign weights to vertices in a graph, there are two typical combinatorial optimization problems called the Price-Collecting Steiner Tree problem (PCST) and the Node-Weighted Steiner Tree problem (NWST). The main difference is that in PCST each node is assigned a benefit value, while in NWST each node is assigned a node cost. Thus, from the min-cost perspective, the total cost of PCST consists of node weights *not being selected*, while the cost of NWST consists of node weights *being selected*.

This slight difference makes a huge gap between them. The PCST problem can be constantly approximated in polynomial time. A typical solution is a primal-dual scheme similar to the Steiner Forest problem. However, for NWST, it has been proved that it is NP-hard to approximate within $(1-\varepsilon)\ln n$ for every $\varepsilon > 0$ [12] and the currently best known polynomial-time approximation ratio is $1.35 \ln n$.

The NWST problem is very similar to our k-Node-Weighted Steiner Tree problem (1). The major difference is that we have the additional constraint to use at least k relay servers.

V. A Polynomial-Time Approximation Scheme for k-Node-weighted Steiner Tree

We give the first PTAS on the k-Node-Weighted Steiner Tree (k-NWST) problem (1) formulated in Sec. III with an approximation ratio of $4\ln n + 1$, which can be further improved to $2.7 \ln n + 1$. Note that Node-Weighted Steiner Tree (NWST) is a special case of k-NWST without the constraint on the minimum number of employed servers. Our proposed algorithm requires a feasible solution of NWST as a subroutine. In the following, we will first discuss an existing PTAS for NWST [13], [14], before presenting our algorithm for k-NWST with approximation ratio analysis.

A. An Existing Approximation Scheme for NWST

The LP-relaxation of problem (1) can be obtained as follows:

$$\begin{array}{lrcll}
\underset{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}}{\text{minimize}} & \multicolumn{4}{l}{\sum_{e\in E} c_e z_e + \sum_{i\in F} f_i y_i + \sum_{i\in F, j\in T} c_{i,j} x_{i,j}} \\
\text{subject to} & \sum_{e\in\delta(N)} z_e & \geq & y_i, & (\forall i \in N \subseteq F) \\
& \sum_{i\in F} x_{i,j} & = & 1, & (\forall j \in T) \\
& x_{i,j} & \leq & y_i, & (\forall i \in F, j \in T) \\
& \sum_{i\in F} y_i & \geq & k, & \\
& x_{i,j} & \geq & 0, & (\forall i \in F, j \in T) \\
& y_i & \geq & 0, & (\forall i \in F) \\
& z_e & \geq & 0. & (\forall e \in E)
\end{array}$$

Note that by $\sum_{i\in F} x_{i,j} = 1$, we can get $x_{i,j} \leq 1$. Also, $x_{i,j} \leq 1$ implies that no optimum solution would have $y_i > 1$ for any server i. Furthermore, $y_i \leq 1$ also implies $z_e \leq 1$. Hence, we can omit these three constraints for any server i and client j.

According to [13], [14], the NWST problem is NP-hard to solve and is even NP-hard to approximate within a factor of $(1-\varepsilon)\ln n$ for every $\varepsilon > 0$. Some near-optimal solutions, which have at most $2\ln n$ approximation ratio, have been proposed based on tree merging and spider decomposition [14]. For the sake of simplicity, we will only briefly describe this algorithm, although the currently best one is proposed by Guha and Kuller in [12], who give a PTAS with an approximation ratio of $1.35 \ln n$.

Initially, each terminal node is a tree by itself, or singleton tree. In each iteration, the algorithm will find a node v in V to merge trees into a larger one by a greedy strategy. The criterion of node selection is to minimize the following ratio:

$$\frac{\text{Opening cost plus sum of distances from } v \text{ to the trees}}{\text{Number of trees}}.$$

Here we denote X the employed relay servers and P the tree connecting relay servers and clients. The source node P and relay servers also form a tree in P, thus we denote P_X as the subtree consisting of S and the selected relay server nodes. The distance from v to a node v' is defined as the total cost of edges in a shortest path between them, and the distance from v to a tree P is defined as the minimum distance from v to node v' in the tree P. Obviously, such choice in each iteration would minimize the average node-to-tree distance. The iteration will end when only one tree is left, and the final tree will be the output. The selection of trees for a node seems to be exponential time consuming; however, there is a simple implementation of each iteration in [14] and we omit it here.

The algorithm uses the shortest path between the node and the selected trees to merge the trees into one, and it will end when there is only one tree.

B. A PTAS for Our Problem: k-NWST

We now propose our method to find an approximation algorithm for k-NWST. We here denote γ as the best approximation ratio we know so far, and algorithm $\mathcal{A}$ as the corresponding algorithm for NWST, which is a subroutine in our algorithm.

Here we briefly describe how we apply Lagrangian relaxation to the additional constraint on the minimum number of servers. We obtain the following problem for each fixed Lagrangian multiplier $\lambda > 0$:

$$\begin{array}{lrcll}\underset{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}}{\text{minimize}} & \multicolumn{4}{l}{\sum_{e\in E} c_e z_e + \sum_{i\in F} f_i y_i + \sum_{i\in F, j\in T} c_{i,j} x_{i,j}} \\ & \multicolumn{4}{l}{+\lambda(\sum_{i\in F} y_i - k)} \\ \text{subject to} & \sum_{e\in\delta(N)} z_e & \geq & y_i, & (\forall i \in N \subseteq F) \\ & \sum_{i\in F} x_{i,j} & = & 1, & (\forall j \in T) \\ & x_{i,j} & \leq & y_i, & (\forall i\in F, j \in T) \\ & x_{i,j} & \geq & 0, & (\forall i\in F, j \in T) \\ & y_i & \geq & 0, & (\forall i \in F) \\ & z_e & \geq & 0. & (\forall e \in E)\end{array} \quad (2)$$

We begin our algorithm by searching for proper Lagrangian multipliers. To be more precise, we maintain an interval $[\lambda_1, \lambda_2]$ such that running $\mathcal{A}$ with λ set to λ_i yields a primal solution spanning k_i vertices, with $k_1 < k < k_2$. By the discussion above, the interval can initially be $[0, \sum_e c_e]$. We then run $\mathcal{A}$ using $\lambda = (\lambda_1 + \lambda_2)/2$. If a tree is returned with k vertices, we are done. If it has more than k vertices, we update λ_2 to be $(\lambda_1+\lambda_2)/2$; otherwise it has less than k vertices and we update λ_1 to this value. Such binary search procedure can find two values of λ with negligible difference $\lambda_1 \approx \lambda_2$, for which we get two trees P_1 and P_2 such that $k_1 < k < k_2$.

The next step is to combine them into a new solution which at least opens k servers. We can do so by a convex combination, i.e., $k = \mu_1 k_1 + \mu_2 k_2$, where

$$\mu_1 = \frac{k_2 - k}{k_2 - k_1},\ \mu_2 = \frac{k - k_1}{k_2 - k_1}.$$

Then, we can follow a similar way as Garg did in [15]. Let $P_2' = P_2 \backslash P_1$ and thus $|P_2'| \geq k_2 - k_1$. From trees P_1 and P_2 we have:

$$C_{OPEN}(X_1) + C_D(P_1) + \lambda(k_1 - k) \leq \gamma OPT \quad (3)$$

$$C_{OPEN}(X_2) + C_D(P_2) + \lambda(k_2 - k) \leq \gamma OPT. \quad (4)$$

Here $C_{OPEN}(X)$ denotes the sum of opening costs of servers in X, and $C_D(P)$ denotes the sum of congestion costs on edges in P. We have

$$\begin{aligned} & \mu_1(C_{OPEN}(X_1) + C_D(P_1) + \lambda(k_1 - k)) \\ & +\mu_2(C_{OPEN}(X_2) + C_D(P_2) + \lambda(k_2 - k)) \\ = \ & \mu_1(C_{OPEN}(X_1) + C_D(P_1)) \\ & +\mu_2(C_{OPEN}(X_2) + C_D(P_2)) \\ = \ & \mu_1(\sum_{i\in X_1} f_i + \sum_{e\in P_1} c_e) + \mu_2(\sum_{i\in X_2} f_i + \sum_{e\in P_2} c_e) \quad (5) \\ \leq \ & \gamma OPT. \quad (6)\end{aligned}$$

If $\mu_2 > 1/2$, we can directly use k_2 servers in X_2, since

$$\begin{aligned}\sum_{i\in X_2} f_i + \sum_{e\in P_2} c_e & \leq 2\mu_2(\sum_{i\in X_2} f_i + \sum_{e\in P_2} c_e) \quad (7) \\ & \leq 2\gamma OPT. \quad (8)\end{aligned}$$

Now we suppose that $\mu_1 > 1/2$ in the subsequent discussion. In this case, the tree P_1 is supplemented by vertices from P_2.

For each edge in P_2', we exchange it for two directed edges of the same cost, one pointing each way. Thus, the total edge cost for P_2 is doubled, while the total node cost is as same. These edges form an Euler tour containing all vertices of P_2' and each vertex appears twice in the tour. Next, from each vertex in P_2, start following the Euler tour in a clockwise direction until $2(k-k_1)$ nodes of P_2 are encountered, including repeats. This gives us at least $2(k_2 - k_1)$ different subpaths of the Euler tour, two for each vertex in P_2, and we only need such subtour with least cost. Note that opening cost are also included, (that is, such cost includes both edge cost along the tour and opening cost for servers in the tour) which is the main difference from Garg's algorithm.

We then add a edge from S to X', denoted as $PATH(S, X')$, to connect nodes in these two sets and get a tree. The cost can be no more than OPT, if we preprocess the graph to throw away all relay servers whose distance to the root is greater than OPT. Finally, all these servers open and for each client j, it can change server from i to i', if $c_{i,j} \geq c_{i',j}$, and we get the final answer.

We now analyze the approximation performance of our algorithm. The average of cost is $(k - k_1)/(k_2 - k_1)$ times the cost of cycle, and the set of $k - k_1$ servers with least cost, denoted as X', cannot exceed this value. That is,

$$\begin{aligned} & C_{OPEN}(X') + C_D(P') \\ \leq \ & \frac{k - k_1}{k_2 - k_1} \cdot 2(C_{OPEN}(X_2) + C_D(P_2)) \quad (9) \\ = \ & 2\mu_2(C_{OPEN}(X_2) + C_D(P_2)). \quad (10)\end{aligned}$$

Here P' is the set of edges interconnecting servers in X' with least cost. The total cost for servers are thus

$$\begin{aligned} & C_D(PATH(S, X')) + C_{OPEN}(X_1) + C_D(P_1) \\ & +C_{OPEN}(X') + C_D(P') \\ \leq \ & C_D(PATH(S, X')) + C_{OPEN}(X_1) + C_D(P_1) \\ & +2\mu_2(C_{OPEN}(X_2) + C_D(P_2)) \\ \leq \ & C_D(PATH(S, X')) + 2\mu_1(C_{OPEN}(X_1) + C_D(P_1)) \\ & +2\mu_2(C_{OPEN}(X_2) + C_D(P_2)). \quad (11)\end{aligned}$$

And we have

$$C_D(PATH(S, X')) \leq OPT. \quad (12)$$

Thus, the total cost is upper bounded as

$$\begin{aligned}
& C_{OPEN}(X) + C_D(X) \\
\leq\ & 2\mu_1(C_{OPEN}(X_1) + C_D(X_1)) \\
& +2\mu_2(C_{OPEN}(X_2) + C_D(X_2)) + C_D(PATH(S, X')) \\
\leq\ & 2\mu_1\gamma OPT + 2\mu_2\gamma OPT + OPT \quad (13) \\
=\ & 2\gamma OPT + OPT \quad (14) \\
=\ & (2\gamma + 1)OPT. \quad (15)
\end{aligned}$$

In summary, by using Lagrangian relaxation, we can achieve a $2\gamma + 1$ approximation to the k-NWST, if the subroutine for NWST has an approximation ratio of γ. Given that the best known approximation ratio for NWST is $1.35 \ln n$, our algorithm can approximate k-NWST with a ratio of $2 \cdot 1.35 \ln n + 1 = 2.7 \ln n + 1$.

VI. Simulations

To evaluate the effectiveness of the proposed min-cost webcast under joint pricing of congestion pricing as well as data and server cost, we conduct simulations based on real-world inter-server and client-server delay traces collected from PlanetLab and from the Seattle project [3].

We monitor the RTTs among 8 Planet nodes for a 15-day period and treat them as the source and CDN edge servers in the simulation. We also monitor the RTTs from the 8 Planet nodes to 19 Seattle nodes, including personal computers and mobile phones on the Seattle platform, and treat them as the clients in our simulation. We choose the median RTT between each pair of nodes as the delay estimate of the pair. The end-to-end delay on a certain path is calculated by summing up the RTTs of all the edges on it divided by 2. Fig. 2 shows the empirical CDF of delays between servers, and the CDF of delays between servers and clients in the traces. We can see that the median delay among servers is about 110 ms, while the median client-server delay is about 1250 ms.

The cost of opening a server includes a per-minute fee on virtualized instances and a fee on data transmission. We synthesize the server opening costs (including data costs) from the pricing policy on Amazon Web Service. [1], where the opening cost of a server is mainly affected by its geographic location. In particular, we set the server cost to $1900, 1900, 1900, 1200, 1200, 2500, 1700, 1700$ with a positive weight in our optimization.

We compare our algorithm with a **Baseline Algorithm**, which randomly employs a subset of servers, with no inter-server transfer, and connects each client to its closet server. Fig. 3 shows the total cost, congestion cost, and server cost computed by our algorithm, and Fig. 4 shows these costs under the baseline algorithm. It is clear that as the number of servers increases, the server opening cost will increase and the congestion cost will drop. However, the total cost does not drop since the opening cost is significant when the webcast operator wishes to use more servers. Note that the minimum total cost under our algorithm is reached when the application opens two servers, while the baseline requires three. Therefore, our algorithm not only saves the total cost, but also saves the computational resources and helps to reduce the traffic.

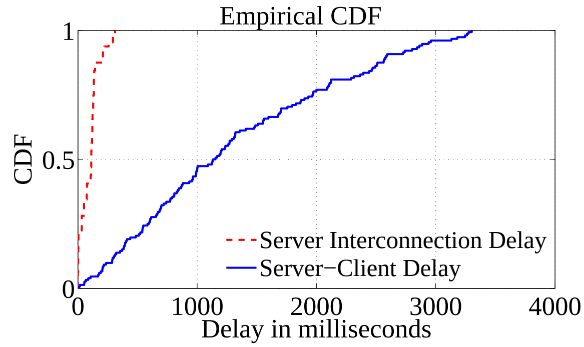

Fig. 2. The CDFs of inter-server RTTs and server-client RTTs.

We further evaluate the mean source-to-client delays achieved under our algorithm and the baseline algorithm in Fig. 5. First, we see that when more CDN servers are used, the delay will be decreased. Moreover, our algorithm also outperforms the baseline algorithm in terms of delay performance, because we explicitly consider the congestion cost, which depends on the latencies in our optimization problem.

VII. Related Work

The concept of responsive pricing for congestion control has existed for a long time. In close-loop feedback pricing [16], the network load, measured in terms of buffer occupancy at the gateway, is converted to a price per packet for users' adaptive applications to decide how much data to transmit. In a study of revenue and welfare maximization for customer calls [17], users initiate calls that have different resource requirements and call duration. Based on the network congestion level, the service provider charges a fee per call, which in turn affects the user demand. Time-dependent usage-based pricing [18] assumes some form of utility functions adopted by customers, and aims to compute the dynamic prices to be offered to customers, using convex optimization, with the objective of minimizing the cost of overusing capacity on bottleneck links and shifting away peak demand. This paper is similar to the above work in that pricing is dependent on network states. In a study of congestion aware pricing for Internet media streaming, [19] use pricing to indirectly control the way that application providers construct their session-specific overlay structures on which to route their traffic, which is similar to the *Space Information Flow* problem [20], [21] under a relay number constraint.

However, instead of using pricing only to improve congestion control, we use pricing to consider cost for transferring data, running virtualized instances, and controling congestion leve as well. Sine the optimal solution is a tree, we can reduce the pricing on bandwidth-delay products to sum of delay in each edge. Under our proposed pricing policy based on congestion fee, data transferring fee, and per-minute pricing on

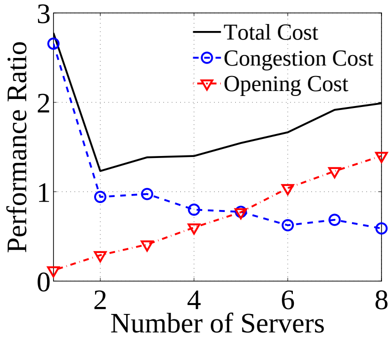

Fig. 3. The cost under our algorithm.

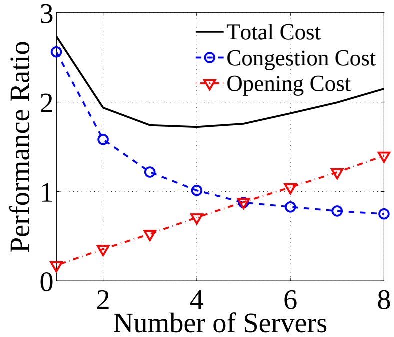

Fig. 4. The cost under the baseline algorithm.

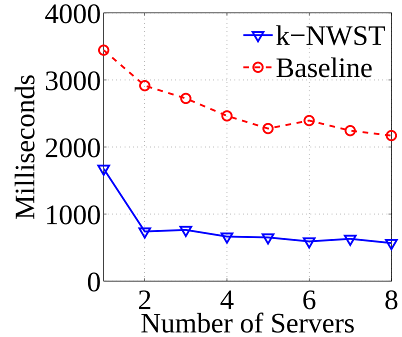

Fig. 5. The mean source-to-client delays.

virtualized instances, every application will be incentivized to minimize its aggregate "waiting data" incurred on the Internet and open less virtualized servers, thus alleviating congestion and data workload. In addition, we also consider cost for data transferring and running virtualized instances.

Our problem is a generalization of Steiner Tree problem, Facility Location problem, and Node-Weighted Steiner Tree problem. All of three are well known NP-hard problems and there are a lot of results about approximation algorithms for them. All existing results of them cannot directly solve the problem when lower bounding the number of open relay servers. Our Lagrangean relaxation can approximate the result with a slight penalty on approximation ratio in polynomial time.

VIII. Concluding Remarks

In this paper, we have proposed a joint pricing policy of data, congestion and virtual servers for live webcast operators as well as an efficient strategy for a webcast operator to compute its min-cost video multicast overlay topology. Assuming webcast operators can employ CDN nodes and datacenters as relay servers in their streaming overlay, we propose to charge a webcast operator for data transfers, server cost, as well as the congestion level incurred on the Internet. We set the congestion fee in proportion to the sum of bandwidth-delay products on all the links in the streaming overlay formed by the webcast operator, to encourage them to form low-cost overlays and thus reduce their impact on the Internet congestion level.

Under such a joint pricing model, we formulate the min-cost webcasting via at least k CDN servers as a problem that we call "k-Node-Weighted Steiner tree" (k-NWST), which is a generalization of several well-known NP-hard problems including the Uncapacitated Facility Location (UFL) problem, the Steiner Tree problem, and the Node-Weighted Steiner Tree (NWST) problem. We give the first polynomial-time approximation scheme to the proposed k-NWST problem with an approximation ratio of $2.7 \ln n + 1$, where n is the number of participating clients.

References

[1] "Amazon Web Services," http://aws.amazon.com/.

[2] M. Vardy, "Telus tuneage: Now powered by rdio," *The Next Web*, http://thenextweb.com/ca/2011/08/03/telus-tuneage-now-powered-by-rdio/.

[3] "Seattle," http://seattle.poly.edu/.

[4] J. Walrand, *Economic Models of Communication Networks*. Springer Publishing Company, 2008.

[5] S. Sen, C. Joe-Wong, S. Ha, and M. Chiang, "Pricing data: A look at past proposals, current plans, and future trends," Tech. Rep., 2012.

[6] D. Frakes, "AT&T announces tethering details and new plans for iPhone, iPad," *InfoWorld*, June 2 2010.

[7] C. Kang, "AT&T wireless scraps flat-rate internet plan," *The Washington Post*, 2010.

[8] T. D. Hau, "Electronic road pricing: Developments in hong kong 1983-1989." *Journal of Transport Economics and Policy*, vol. 24, no. 2, pp. 203–214, May 1990.

[9] J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanità, "An improved lp-based approximation for steiner tree," in *Proceedings of the 42nd ACM symposium on Theory of computing*. ACM, 2010, pp. 583–592.

[10] M. Chlebík and J. Chlebíková, "The steiner tree problem on graphs: Inapproximability results," *Theoretical Computer Science*, vol. 406, no. 3, pp. 207–214, 2008.

[11] S. Li, "A 1.488 approximation algorithm for the uncapacitated facility location problem," *Information and Computation*, vol. 222, pp. 45–58, 2013.

[12] S. Guha and S. Khuller, "Improved methods for approximating node weighted steiner trees and connected dominating sets," *Information and Computation*, vol. 150, no. 1, pp. 57 – 74, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0890540198927547

[13] C. Lund and M. Yannakakis, "On the hardness of approximating minimization problems," *Journal of the ACM (JACM)*, vol. 41, no. 5, pp. 960–981, 1994.

[14] P. Klein and R. Ravi, "A nearly best-possible approximation algorithm for node-weighted steiner trees," *J. Algorithms*, vol. 19, no. 1, pp. 104–115, Jul. 1995. [Online]. Available: http://dx.doi.org/10.1006/jagm.1995.1029

[15] N. Garg, "A 3-approximation for the minimum tree spanning k vertices," in *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*, Oct 1996, pp. 302–309.

[16] J. Murphy and L. Murphy, "Bandwidth allocation by pricing in ATM networks," *IFIP Trans. C: Communications Systems*, vol. C, no. 24, pp. 333–351, 1994.

[17] I. C. Paschalidis and J. N. Tsitsikilis, "Congestion-dependent pricing of network services," *IEEE/ACM Transactions on Networking*, no. 8, pp. 171–184, 1998.

[18] C. Joe-Wong, S. Ha, and M. Chiang, "Time-dependent broadband pricing: Feasibility and benefits," in *Proc. of ICDCS*, June 2011.

[19] D. Niu and B. Li, "Congestion-aware internet pricing for media streaming," in *Proc. of the 3rd Workshop on Smart Data Pricing*, Toronto, Canada, May 2 2014.

[20] J. Huang, X. Yin, X. Zhang, X. Du, and Z. Li, "On space information flow: Single multicast," in *Proc. the International Symposium on Network Coding (NetCod)*, 2013.

[21] Y. Hu, D. Niu, and Z. Li, "Internet video multicast via constrained space information flow," *IEEE MMTC E-letter*, vol. 9, no. 3, April 2014.