Brief Announcement: Live Streaming with Utilities, Quality and Cost

Ymir Vigfusson
Reykjavik University
Reykjavik, Iceland, 105

Ken Birman
Cornell University
Ithaca, New York, 14850

Daniel A. Freedman
Cornell University
Ithaca, New York, 14850

Qi Huang
Cornell University
Ithaca, New York, 14850

Kristján V. Jónsson
Reykjavik University
Reykjavik, Iceland, 105

Gunnar Sigurbjörnsson
Reykjavik University
Reykjavik, Iceland, 105

Categories and Subject Descriptors

C.2.4 [**Computer Communication**]: Distributed Systems

General Terms

Algorithms, Theory

Keywords

live streaming, utility, optimization, approximation algorithm

1. INTRODUCTION

The growth in Internet traffic associated with video streaming and sharing of live video content is so rapid that it may soon dwarf all other forms of Internet content. By late 2012, Internet video alone is projected to generate almost 10 exabytes of traffic per month, accounting for nearly 50 percent of all Internet traffic [3]. ISPs and content providers are faced with the challenge of devising and deploying technologies to accommodate the surging demand for bandwidth. Data generated in real-time such as by live video broadcasts (e.g. sports games or new episodes of popular TV shows), chat systems, immersive virtual reality applications and games typically can't be cached at all. In today's systems, each client may pull such information on its own point-to-point stream directly from the data center, even if large numbers of clients share interest in at least some aspects of the data.

Here, we lay the groundwork for a new overlay networking architecture called GRADIENT aimed at reducing the load on providers of live-streaming content. At the crux of GRADIENT is an algorithm to construct dissemination overlays for each data stream. Nodes express their utility for receiving each stream at a given rate, which allows us to explore the trade-off between offering a lower-quality stream to a greater number of nodes and high-quality transmissions for fewer nodes. The idea is that intermediate nodes can downgrade the quality of the streams they receive and transmit at a lower rate if needed.

We present a cost model of the network and users, and give an effective algorithm to route streams to balance user utility with bandwidth costs by transforming inflight data to match the live stream to the preferences and requirements of the consumer.

2. MODEL AND ALGORITHM

PODC'12, July 16–18, 2012, Madeira, Portugal.
ACM 978-1-4503-1450-3/12/07.

Consider a collection $\mathcal{S}$ of content streams that must be disseminated over an undirected graph $G = (V, E)$. Each edge $e \in E$ has a *cost* $c_e \geq 0$, reflecting e.g. actual unit bandwidth costs. For simplicity, we assume that the source streams originate at a single, abstract *source* node $s \in V$. This assumption is reasonable in our context since services must store media contents (such as a CDN) or maintain consistency (such as a virtual reality service) at some central location, but is not restrictive since one can model multiple content sources by connecting each to s at a zero cost.

Other nodes $v \in V$ subscribe to a subset of the streams in $\mathcal{S}$, and express preferences for the quality they receive for each stream in terms of a *utility* function. Note that some nodes in G need not be subscribed to any stream, but may instead act as proxies. In our scenarios, nodes tend to subscribe to multiple concurrent streams, such as different object update streams in the case of virtual reality, or media in the case where a node collectively represents the customers of an ISP. We express these subscriptions in terms of utility: each node $v \in V$ derives $u_i(v, r) \geq 0$ utility for receiving stream $i \in \mathcal{S}$ at rate r. For convenience, we assume that each rate r is chosen among finitely many rates $0 = r_0 < r_1 < \cdots < r_k$, and $u_i(v, 0) = 0$ always. For example, $(r_0, r_1, r_2) = (0, 200\text{ Kbps}, 400\text{ Kbps})$ means that stream i may be received by subscribers at either 200 or 400 Kbps, or not at all. Let $R = \{r_0, r_1, \ldots, r_k\}$.

Users receive zero utility, $u_i(i, \cdot) = 0$, if they are not interested in stream $i \in \mathcal{S}$. We further assume that utility grows monotonically in r, more specifically that receiving a stream i at rate $r_a > r_b$ provides more benefit for the user so $u_i(v, r_a) \geq u_i(v, r_b)$. Note that if a stream is not available at a high rate r_j then $u_i(v, r_j) = u_i(v, r_{j-1})$, i.e., the *marginal* utility is zero.

High-Level Goal. We define a *routing tree* here to be a directed tree $T \subseteq E$ rooted away from s, along with rates $\rho(T, e)$ for $e \in T$ such that rates along a directed path from s are non-increasing. With abuse of notation, for each $e \in T$ we let $\rho(T, v) = \rho(T, (u, v))$ be the incoming rate to vertex v in the tree. Set $\rho(T, v) = 0$ and $\rho(T, e) = 0$ for vertices v and edges e not in T.

At a high level, our goal is to find a collection of routing trees T_i in G for every stream $i \in \mathcal{S}$ to maximize the utilities of nodes who receive each stream, while simultaneously minimizing the pro-rated cost of the trees. More specifically, we wish to find a collection of trees $(T_i)_{i \in \mathcal{S}}$ to maximize

$$\sum_{i \in \mathcal{S}} \sum_{v \in V} u_i(v, \rho(T_i, v)) - \sum_{i \in \mathcal{S}} \sum_{e \in E} \rho(T_i, e) \cdot c_e. \qquad (1)$$

The problem is clearly NP-complete since it generalizes the Steiner tree problem, which corresponds to unit rates and infinite utilities at terminal nodes.

Linear Program. We next formulate the optimization problem

Algorithm 1 Primal-dual approximation algorithm LS.

Input: A graph $G = (V, E)$, edge costs c_e for $e \in E$, streams $\mathcal{S}$ originating in $s \in V$, utility $u_i(v, r) \geq 0$ for node $v \in V$ receiving stream $i \in \mathcal{S}$ at rate $r \in R$ where R is a finite set of possible rates. We augment the graph G as described to produce $G' = (V', E')$ and $\pi_v^r \geq 0$ for $v \in V', r \in R$.

Output: A routing tree over G' for each stream $i \in \mathcal{S}$.

We run the remaining steps for each stream $i \in \mathcal{S}$.

for each rate $r = r_1, r_2, \ldots, r_k$ in R **do**

- Let $C(r) \leftarrow \{\{v\} : v \in V'\}$ be a spanning forest.
- Grow y_S^r uniformly for each untagged component $S \in C(r)$ with $r_S = r$ until either dual inequality is tight.
- If inequality (3) is tight due to an edge e connecting two distinct components in $C(r)$, we merge the components spanned by e in $C(r)$ and tag e.
- If inequality (4) is tight, we tag the component $S \in C(r)$ which we intend to exclude from the graph.
- Stop growing when there are no untagged components $S \in C(r)$ with $r_S = r$ left.

end for

Traverse the list of tagged edges and components in reverse order, discarding items whose removal produce a feasible solution.

above as a linear program. Because the routing trees are independent from one another in our formulation, we will hereafter focus our attention on computing the best routing tree for a single, fixed stream $i \in \mathcal{S}$. The routing trees for each stream can then be composed. We note that the resulting network could place burden on individual users. We defer link capacity concerns since we believe live streams will not represent a bandwidth bottleneck between ISPs, but note that methods from Steiner tree packing [6] may help to minimize the maximum congestion on network edges.

Augmented Graph. For the sake of analysis, it is convenient for each vertex $v \in V$ to demand the stream at a particular rate or not to demand it at all. To accomplish this, we transform the original graph G as follows. Replace each node $v \in V$ with interest in stream i with a chain of nodes $v_0, \ldots, v_k$ and zero cost edges between (v_j, v_{j+1}) for $0 \leq j < k$, such that the original neighbors of v connect to v_0. Node v_j demands stream i at rate r_j with a *prize* $\pi_{v_j, r_j} = u_i(v, r_j)$ for $1 \leq j < k$ and $\pi_{v_0, r_0} = 0$. We further modify the graph by replicating each edge $e \in E$ to create $k + 1$ parallel edges $(e, r_0), (e, r_1), \ldots, (e, r_k) \in E'$ of cost c_e. Let $G' = (V, E')$ denote the final modified graph.

We define r_S as the maximum rate demanded by the vertices $v \in S \subseteq V$ in G', specifically the highest rate which has a non-zero prize in S.

The problem we have been describing is equivalent to the following integer program [1].

$$\text{Min} \sum_{(e,r) \in E'} x_{e,r} \cdot r \cdot c_e + \sum_{\substack{T \subseteq V - \{s\} \\ r \in R}} z_{T,r} \sum_{v \in T} \pi_{v,r} \tag{2}$$

$$\sum_{\substack{(e,r) \in \delta(S) \\ r = r_S}} x_{e,r} + \frac{1}{2} \sum_{\substack{(e,r) \in \delta(S) \\ r > r_S}} x_{e,r} + \sum_{\substack{T \supseteq S \\ r = r_S}} z_{T,r} \geq 1 \; \forall S \subseteq V - \{s\}$$

$$x_{e,r}, z_{T,r} \in \{0, 1\}, \; \forall T \subseteq V - \{s\}, r \in R$$

Here, $\delta(S)$ denotes the set of edges crossing the $(S, V - S)$ cut, i.e. the edges with one endpoint in S and the other in $V - S$. The binary vector $\vec{x}$ corresponds to the edges and edge rates picked as part of the routing tree, and for which we pay a cost. Conversely, the $\vec{z}$ denotes the vertices outside of the routing tree, and for which we pay a penalty equal to the prizes we did not collect. Note that the new cost function (2) is equivalent to the original cost function (1), shifted by the total available prizes. The $\frac{1}{2}$-term allows traffic to be tunneled via a component (requiring at least a pair of edges) at a faster rate than required by the nodes in the component.

We relax the integrality constraints: $x_{e,r}, z_{T,r} \geq 0$ for all $T \subseteq V - \{s\}$ and $r \in R$. The dual of the linear program is as follows.

$$\text{Max} \sum_{S \subseteq V - \{s\}} y_S$$

$$\sum_{\substack{S: e \in \delta(S) \\ r_S = r}} y_S + \frac{1}{2} \sum_{\substack{S: e \in \delta(S) \\ r_S < r}} y_S \leq r \cdot c_e \quad \forall (e, r) \in E' \tag{3}$$

$$\sum_{\substack{S \subseteq T \\ r = r_S}} y_S \leq \sum_{v \in T} \pi_{v,r} \quad r \in R, \quad \forall T \subseteq V - \{s\} \tag{4}$$

$$y_S \geq 0 \quad \forall S \subseteq V - \{s\}.$$

THEOREM 1. *The solution found by the algorithm in Fig. 1 for stream $i \in \mathcal{S}$ costs at most $5.986 \cdot \texttt{OPT}$.*

PROOF. *(Sketch)* The proof has four steps. We first use randomized doubling [2] to round each traffic rate r in an instance of the problem to $a^{\gamma + j} \geq r$ for a fixed α, value of $\gamma \in [0, 1]$ was uniformly at random, and the lowest integral value of j. The cost of the rounded instance is at most α factor greater than the original.

We next bound the cost of the solution found by the algorithm on the rounded instance as a multiple of the dual solution, since the value of any dual feasible solution is at most the value of the optimum solution for the primal. By looking at the edges (e, r) chosen by the algorithm when the constraint (3) is tight, we find that $\sum_{S: e \in \delta(S); r_S = r_j} y_S \geq c_e \left(r_j - \frac{r_j}{a} \cdot \frac{2\alpha}{2\alpha - 1} \right) = c_e r_j \frac{2\alpha - 3}{2\alpha - 1}$ for $0 \leq j \leq k$. One can derive a similar bound on the cost of the graph components T included by the algorithm.

Next, we analyze LS in the Prize-Collecting Steiner Tree (PCST) framework [4, 5]. If $C(r)$ is the set of active components in the output with $r_S = r$ for $S \in C(r)$, then the number of edges of rate r between the components in $C(r)$ is at most $2C(r)$. Using PCST arguments, the solution found by LS costs at most

$$2 \cdot \tfrac{2\alpha - 1}{2\alpha - 3} \textstyle\sum_{S \subseteq V - \{s\}} y_S \leq \tfrac{4\alpha - 2}{2\alpha - 3} \texttt{OPT} \tag{5}$$

where OPT is the cost of the optimal solution to the linear program of the rounded instance.

Finally, we determine the ideal value of α to minimize rounding error using calculus. Combined with (5), LS produces a solution with cost within a $\frac{(4a-2)(a-1)}{(2a-3)\ln a}$ factor of the optimum. This expression is minimized numerically at $a = 3.447$, yielding a 5.986-approximation algorithm. □

ACKNOWLEDGMENT

We gratefully acknowledge Graduate Studies Grant #080520008 from the Icelandic Centre for Research (Rannís).

3. REFERENCES

[1] G. Calinescu, C. Fernandes, I. Mandoiu, A. Olshevsky, K. Yang, and A. Zelikovsky. Primal-dual algorithms for QoS multimedia multicast. In *Proceedings. of IEEE GLOBECOM*, pages 1–17, 2003.

[2] M. Charikar, J. S. Naor, and B. Schieber. Resource optimization in QoS multicast routing of real-time multimedia. *IEEE/ACM Trans. Netw.*, 12(2):340–348, 2004.

[3] Cisco. Approaching the zettabyte Era. *Cisco Visual Networking Index*, page 23, 2008.

[4] M. X. Goemans and D. P. Williamson. A general approximation technique for constrained forest problems. In *SODA '92: Proceedings of the third annual ACM-SIAM symposium on Discrete algorithms*, pages 307–316, Philadelphia, PA, USA, 1992. Society for Industrial and Applied Mathematics.

[5] M. X. Goemans and D. P. Williamson. The primal-dual method for approximation algorithms and its application to network design problems. pages 144–191, 1997.

[6] K. Jain, M. Mahdian, and M. R. Salavatipour. Packing Steiner trees. In *SODA '03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 266–274, Philadelphia, PA, USA, 2003. Society for Industrial and Applied Mathematics.