Optimal Resource Allocation in Overlay Multicast *

Yi Cui, Yuan Xue, Klara Nahrstedt
Department of Computer Science, University of Illinois at Urbana-Champaign
{*yicui, xue, klara*}@cs.uiuc.edu

Abstract

This paper targets the problem of optimal resource allocation in overlay multicast, which poses both theoretical and practical challenges. Theoretically, resource allocation among overlay flows is not only subject to the network capacity constraint, but also the data constraint, mainly due to the dual role of end hosts as both receivers and senders. Practically, existing distributed resource allocation schemes assume the network links to be capable of measuring flow rates, calculating and communicating price signals, none of which actually exists in the Internet today. We address these challenges as follows. First, we formalize the problem using nonlinear optimization theory, which incorporates both network constraint and data constraint. Based on our theoretical framework, we propose a distributed algorithm, which is proved to converge to the optimal point, where the aggregate utility of all receivers is maximized. Second, we propose an end-host-based solution, which relies on the coordination of end hosts to accomplish tasks originally assigned to network links. Our solution can be directly deployed without any changes to the existing network infrastructure.

1 Introduction

Multicast is an important communication paradigm, upon which many network applications can be built, such as teleconferencing, multimedia distribution, etc. Recently, overlay multicast[13] has gained intensive consideration, for the following advantages it offers. First, it works around the deficiency of infrastructure support for multicast, i.e., IP multicast is largely unavailable in the Internet. Instead, by organizing end hosts into an overlay network, multicast can be achieved through data relay among end hosts over unicast. Second, it accommodates the network heterogeneity by supporting multi-rate multicast, where receivers in the same group can retrieve service at different rates. In IP-multicast-based solutions, this is mainly achieved by layered streaming[11], in which a stream is encoded into multiple layers and fed into different multicast channels. A receiver only needs a subset of them to recover the stream with certain quality degradation. However, the receiver can only choose from a discrete set of streaming rates. On the other hand, overlay multicast addresses this problem with greater flexibility. Besides the layered approach[14], all end-to-end stream adaptation techniques (frame dropping, transcoding[4], etc.) can be applied, since the data relay happens on each end host, which offers more functionalities than simply forwarding packets. In this way, the receiver is allowed to choose its streaming rate on a continuous range.

This paper targets the problem of optimal resource allocation in overlay multicast. An optimal solution should maximize the aggregate utilities of all receivers, subject to various constraints, such as the network link capacity. Here, the receiver utility is defined as a function of the receiver's streaming rate. The function value can be understood as the perceived quality, user satisfaction, etc. Meanwhile, various fairness objectives (max-min, proportional, etc.) can be achieved when we choose appropriate utility functions for receivers[5][6].

Utility-based resource allocation has been explored in the rate control of unicast[10][6] and IP multicast[8][7]. In these solutions, a "price" is associated with each individual network link. The link iteratively updates its price based on the aggregate rate of flows going through it. The receiver in turn collects the prices of all links on its unicast/multicast path and calculates the overall network price. Then, it adjusts the streaming rate such that its "net benefit", the receiver utility minus the network cost, is maximized. It is shown that this iterative algorithm converges to the optimal point, where aggregate utility of all receivers is maximized. Although using similar approach, we show that resource allocation in overlay multicast faces unique challenges both theoretically and practically, making this problem a com-

*This work was supported by NSF CISE infrastructure grant under contract number NSF EIA 99-72884, and DoD Multi-disciplinary University Research Initiative (MURI) program administered by the Office of Naval Research under Grant NAVY CU 37515-6281. Any opinions, findings, and conclusions are those of the authors and do not necessarily reflect the views of the above agencies.

pletely different one to which none of the past solutions can be applied.

Theoretically, resource allocation in overlay multicast is not only subject to the network capacity constraint, but also the data constraint on the relaying node. This is mainly due to the dual role of end hosts as both receivers and senders. Obviously, a receiver cannot relay the stream to its downstream receiver at a rate higher than its own receiving rate. This issue never arises in the context of unicast or IP multicast, where the receiver is always the sink of a unicast/multicast path. An example can be found at Sec. 3.4 to justify our argument.

Practically, existing solutions[10][7] require the network link (actually the router connected to it) to be capable of measuring flow rates, calculating link price and communicating price signal, none of which exists in the Internet today. In fact, they are against the initial design objective of overlay network, which is to avoid any change to the existing infrastructure by migrating the required functionalities to the end hosts. In accordance with the same objective, a practical solution should purely depends on the coordination of end hosts.

The main purpose of our work is to address the above challenges. Our contributions are as follows.

On theoretical challenge, we model the overlay resource allocation problem using nonlinear optimization theory. Our formalization incorporates not only the network constraint proposed in previous works such as[10], but also the data constraint. We address this constraint by pricing the data relay in overlay multicast, i.e., a receiver has to pay its parent for relaying the stream. We propose a distributed algorithm, where each overlay flow adjusts its rate according to both its network price and its "data price". We prove that the rate allocation converges to the optimal point, at which the aggregate utility of all receivers is maximized.

On practical challenge, we propose an end-host-based solution, where the tasks originally assigned to the network links and overlay flows are handled by end hosts. Two protocols, *Link Delegation* and *Aggregate Route Pricing*, are presented. Both of them rely on the coordination of end hosts to calculate and exchange network/data price signals, and adjust the flow rate. In contrast to past solutions[10][7], our solution can be deployed to overlay multicast without any change to the existing infrastructure.

The remainder of this paper is organized as follows. Sec. 2 introduces the network model. Sec. 3 presents the problem formulation and proposes a distributed algorithm. Sec. 4 discusses the protocol design and implementation in overlay network environment. Finally, we show experimental results in Sec. 5, discuss the related work in Sec. 6, and conclude in Sec. 7.

2 Network Model

We consider an overlay network consisting of H end hosts, denoted as $\mathcal{H} = \{1, 2, \ldots, H\}$. One end host is the server, hence the source of the multicast session. Other end hosts relay the multicast stream via unicast in a peer-to-peer fashion. The multicast session consists of F unicast end-to-end flows ($F = H - 1$), denoted as $\mathcal{F} = \{1, 2, \ldots, F\}$. Each flow $f \in \mathcal{F}$ has a rate x_f. We define a rate vector $\boldsymbol{x} = (x_f, f \in \mathcal{F})$. If a host is the destination of a flow f and the source of another flow f', then f' is the child flow of f, denoted as $f \rightarrow f'$. Likewise, if the source of f and the destination of f^p turns out to be one host, then f^p is the parent flow of f, denoted as $f^p \rightarrow f$.

Let us suppose that the overlay network consists of L physical network links, denoted as $\mathcal{L} = \{1, 2, ..., L\}$. The bandwidth capacity of each link $l \in \mathcal{L}$ is c_l. We collect them into a link capacity vector $\boldsymbol{c} = (c_l, l \in \mathcal{L})$. Each flow f passes a subset of physical network links, denoted as $\mathcal{L}(f) \subseteq \mathcal{L}$. For each link l, $\mathcal{F}(l) = \{f \in \mathcal{F} \mid l \in \mathcal{L}(f)\}$ is the set of flows that pass through it.

Now, we define a $L \times F$ matrix $\boldsymbol{A}$. $A_{lf} = 1$, if flow f goes through the link l, i.e., $f \in \mathcal{F}(l)$. Otherwise, $A_{lf} = 0$. $\boldsymbol{A}$ gives the physical network resource usage pattern of an overlay network. It follows that the sum rate of all flows that go through the link l should not exceed its capacity c_l. Formally, such capacity constraint is expressed as follows.

$$\boldsymbol{A} \cdot \boldsymbol{x} \leq \boldsymbol{c} \tag{1}$$

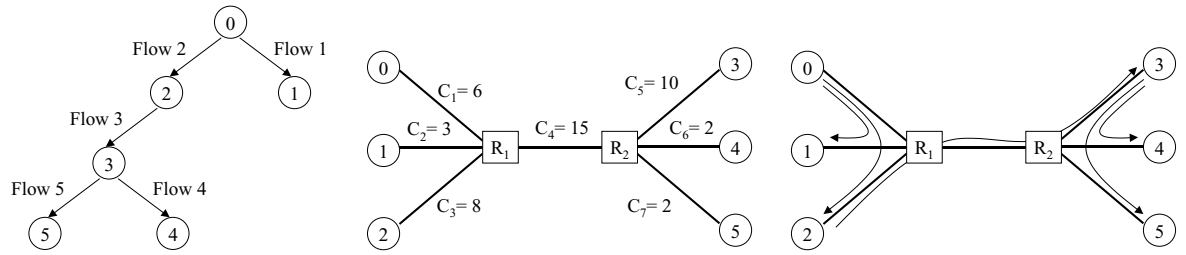
Moreover, the data constraint of overlay multicast states that a host can not relay the stream to its downstream host at a rate higher than its receiving rate, i.e., a flow's rate can not exceed its parent flow's rate, if it has one. Formally, if $f \rightarrow f'$, then $x_{f'} \leq x_f$. We define a $F \times F$ matrix $\boldsymbol{B}$ as follows.

$$B_{f'f} = \begin{cases} -1 & \text{if } f \rightarrow f' \\ 1 & \text{if } f' = f \text{ and } f \text{ has a parent flow} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$\boldsymbol{B}$ specifies the relaying relationship and data dependency in overlay multicast. It is determined by the topology of the overlay multicast tree[13]. Hence, the data constraint can be formalized as follows.

$$\boldsymbol{B} \cdot \boldsymbol{x} \leq \boldsymbol{0} \tag{3}$$

We collect above notations into Tab. 2. In the example by Fig. 1, there are 5 overlay multicast flows ($F = 5$). The physical network consists of 7 links ($L = 7$). Hence, In-



(a) Overlay Multicast (b) Physical Network Topologe (c) Overlay Multicast on the Physical Network

Figure 1. Sample Illustrating Overlay Multicast

equality (1) becomes

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} \leq \begin{pmatrix} 6 \\ 3 \\ 8 \\ 15 \\ 10 \\ 2 \\ 2 \end{pmatrix}$$

Inequality (3) becomes

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} \leq \mathbf{0}$$

Notation	Definition
$h \in \mathcal{H} = \{1, \ldots, H\}$	End Host
$f \in \mathcal{F} = \{1, \ldots, F\}$	Unicast Flow in Overlay Multicast
$\boldsymbol{x} = (x_f, f \in \mathcal{F})$	Flow Rate of $f \in \mathcal{F}$
$l \in \mathcal{L} = \{1, \ldots, L\}$	Physical Network Link
$\boldsymbol{c} = (c_l, l \in \mathcal{L})$	Link Capacity of $l \in \mathcal{L}$
$f \rightarrow f'$	f' is the Child Flow of f
$f^p \rightarrow f$	f^p is the Parent Flow of f
$\mathcal{L}(f) \subseteq \mathcal{L}$	Set of Links that f Goes Through
$\mathcal{F}(l) \subseteq \mathcal{F}$	Set of Flows that Go Through l
$\boldsymbol{A} = (A_{lf})_{L \times F}$	Link Capacity Constraint Matrix
$\boldsymbol{B} = (B_{f'f})_{F \times F}$	Data Constraint Matrix

Table 1. Notations in Sec. 2

Throughout the paper, we will use this example to illustrate our algorithm and protocol.

3 Optimal resource allocation

3.1 Problem Formulation

We associate each flow (or a receiver) $f \in \mathcal{F}$ with an *utility function* $U_f(x_f) : \Re_+ \rightarrow \Re_+$. We make the following assumptions about U_f.

- **A1**. On the interval $I_f = [m_f, M_f]$, the utility functions U_f are increasing, strictly concave and twice continuously differentiable.
- **A2**. The curvatures of U_f are bounded away from zero on I_f: $-U_f''(x_f) \geq 1/\kappa_f > 0$
- **A3**. U_f is additive so that the aggregated utility of rate allocation $\boldsymbol{x} = (x_f, f \in \mathcal{F})$ is $\sum_{f \in \mathcal{F}} U_f(x_f)$.

We investigate the optimal rate allocation in the sense of maximizing the aggregated utility function. We now formulate the problem of optimal resource allocation in an overlay network as the following constrained non-linear optimization problem.

$$\begin{aligned} \mathbf{P}: \quad & \textbf{maximize} && \sum_{f \in \mathcal{F}} U_f(x_f) && (4) \\ & \textbf{subject to} && \boldsymbol{A} \cdot \boldsymbol{x} \leq \boldsymbol{c} && (5) \\ & && \boldsymbol{B} \cdot \boldsymbol{x} \leq \mathbf{0} && (6) \\ & \textbf{over} && \boldsymbol{x} \in I_f && (7) \end{aligned}$$

By Assumption **A1**, objective function (4) is differentiable and strictly concave. Also, the feasible region of constraints (5) and (6) is compact. By non-linear optimization theory, there exists a maximizing value of argument $\boldsymbol{x}$ for the above optimization problem. Let us consider the Lagrangian form of this optimization problem:

$$\begin{aligned} & L(\boldsymbol{x}, \boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) && (8) \\ = & \sum_{f \in \mathcal{F}} U_f(x_f) - \boldsymbol{\mu}^\alpha (\boldsymbol{A} \cdot \boldsymbol{x} - \boldsymbol{c}) - \boldsymbol{\mu}^\beta (\boldsymbol{B} \cdot \boldsymbol{x}) \\ = & \sum_{f \in \mathcal{F}} U_f(x_f) - \sum_{f \in \mathcal{F}} x_f \sum_{l \in \mathcal{L}} \mu_l^\alpha A_{lf} \\ & - \sum_{f \in \mathcal{F}} x_f \sum_{f' \in \mathcal{F}} \mu_{f'}^\beta B_{f'f} + \sum_{l \in \mathcal{F}} \mu_l^\alpha c_l \end{aligned}$$

where $\boldsymbol{\mu}^\alpha = (\mu_l^\alpha, l \in \mathcal{L})$ and $\boldsymbol{\mu}^\beta = (\mu_f^\beta, f \in \mathcal{F})$ are vectors of Lagrangian multipliers. Eq. (9) can be further derived as follows.

We then define two new vectors $\boldsymbol{\lambda}^\alpha = (\lambda_f^\alpha, f \in \mathcal{F})$ and $\boldsymbol{\lambda}^\beta = (\lambda_f^\beta, f \in \mathcal{F})$ as follows.

$$\lambda_f^\alpha = \sum_{l \in \mathcal{L}} \mu_l^\alpha A_{lf} = \sum_{l \in \mathcal{L}(f)} \mu_l^\alpha \quad (9)$$

$$\lambda_f^\beta = \sum_{f' \in \mathcal{F}} \mu_{f'}^\beta B_{f'f} = \mu_f^\beta - \sum_{f \to f'} \mu_{f'}^\beta \quad (10)$$

Now Eq. (8) becomes

$$\begin{aligned} & L(\boldsymbol{x}, \boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) \\ = & \sum_{f \in \mathcal{F}} U_f(x_f) - \sum_{f \in \mathcal{F}} x_f(\lambda_f^\alpha + \lambda_f^\beta) + \sum_{l \in \mathcal{L}} \mu_l^\alpha c_l \end{aligned}$$

For $\boldsymbol{\mu}^\alpha$, μ_l^α can be understood as the *link price* of l. Consequently, for $\boldsymbol{\lambda}^\alpha$, λ_f^α (Eq. (9)) is the sum of prices of all links that f goes through, or in other words, the *network price* that f has to pay. These two vectors correspond to the network constraint stated in (5).

For $\boldsymbol{\mu}^\beta$, μ_f^β is the *relay price* that f must pay its parent flow f^p for relaying data to f. If f has no parent flow, then $\mu_f^\beta = 0$. Meanwhile, for f^p, μ_f^β can be understood as its *relay benefit* for doing so. Now for $\boldsymbol{\lambda}^\beta$, we can interpretate λ_f^β (Eq. (10)) as f's *data price*, which is the difference of f's relay price μ_f^β and its relay benefit from all its children $\sum_{f \to f'} \mu_{f'}^\beta$. There are four cases:

1. f has both parent and children (flow 3 in Fig. 1).
2. f has parent but no children (flows 4 and 5 in Fig. 1), where $\sum_{f \to f'} \mu_{f'}^\beta = 0$.
3. f has no parent but children (flow 2 in Fig. 1), where $\mu_f^\beta = 0$.
4. f has neither parent nor children (flow 1 in Fig. 1), where $\lambda_f^\beta = 0$.

In summary, $\boldsymbol{\mu}^\beta$ and $\boldsymbol{\lambda}^\beta$ correspond to the data constraint stated in (6).

3.2 Dual Problem

Solving the objective function (4) requires global coordination of all flows, which is impractical in distributed environment such as the overlay network. In order to achieve a distributed solution, we first look at the dual problem of $\mathbf{P}$ as follows.

$$\mathbf{D}: \min_{\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta \geq \mathbf{0}} D(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) \quad (11)$$

where

$$D(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta)$$

Notation	Definition
$U_f(x_f)$ $(f \in \mathcal{F})$	Utility Function of x_f
$I_f = [m_f, M_f]$	Feasible Range of $U_f(x_f)$
$\boldsymbol{\mu}^\alpha = (\mu_l^\alpha, l \in \mathcal{L})$	Link Price of l
$\boldsymbol{\mu}^\beta = (\mu_f^\beta, f \in \mathcal{F})$	Relay Price for f
$\boldsymbol{\lambda}^\alpha = (\lambda_f^\alpha, f \in \mathcal{F})$	Network Price for f
$\boldsymbol{\lambda}^\beta = (\lambda_f^\beta, f \in \mathcal{F})$	Data Price for f
$\Phi(x_f)$ $(f \in \mathcal{F})$	Net Benefit of f
γ	Step Size
$x_f(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta)$ $(f \in \mathcal{F})$	Rate Adaptation Function of x_f
$[x]_m^M$ (x is any variable)	$\min\{\max\{x, M\}, m\}$
$[x]^+$ (x is any variable)	$\max\{x, 0\}$

Table 2. Notations in Sec. 3

$$\begin{aligned} & = \max_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) \\ & = \max_{\boldsymbol{x}} \sum_{f \in \mathcal{F}} \underbrace{(U_f(x_f) - (\lambda_f^\alpha + \lambda_f^\beta)x_f)}_{\Phi(x_f)} + \sum_{l \in \mathcal{L}} \mu_l^\alpha c_l \quad (12) \end{aligned}$$

Since λ_f^α and λ_f^β are respectively the network price and data price of f, it is clear that $(\lambda_f^\alpha + \lambda_f^\beta)x_f$ is the *overall cost* for f. Then $\Phi(x_f)$ is f's "net benefit", i.e., the difference of its utility and cost. By the separation nature of Lagrangian form, maximizing $L(\boldsymbol{x}, \boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta)$ can be decomposed into separately maximizing $\Phi(x_f)$ for each flow $f \in \mathcal{F}$ (Sec. 3.4.2 in [3]). Now we have

$$D(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) = \sum_{f \in \mathcal{F}} \max_{x_f \in I_f} \{\Phi(x_f)\} + \sum_{l \in \mathcal{L}} \mu_l^\alpha c_l \quad (13)$$

By Assumption **A1**, U_f is strictly concave and twice continuously differentiable. Therefore, a unique maximizer of $\Phi(x_f)$ exists when

$$\frac{d\Phi(x_f)}{dx_f} = U_f'(x_f) - (\lambda_f^\alpha + \lambda_f^\beta) = 0$$

We define the maximizer as below

$$x_f(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) = \arg\max_{x_f \in I_f} \{\Phi(x_f)\} = [U_f'^{-1}(\lambda_f^\alpha + \lambda_f^\beta)]_{m_f}^{M_f} \quad (14)$$

3.3 Algorithm

We solve the dual problem $\mathbf{D}$ using gradient projection method[3]. In this method, $\boldsymbol{\mu}^\alpha$ and $\boldsymbol{\mu}^\beta$ are adjusted in opposite direction to the gradient $\nabla D(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta)$:

$$\mu_l^\alpha(t+1) = [\mu_l^\alpha(t) - \gamma \frac{\partial D(\boldsymbol{\mu}^\alpha(t), \boldsymbol{\mu}^\beta(t))}{\partial \mu_l^\alpha}]^+ \quad (15)$$

$$\mu_f^\beta(t+1) = [\mu_f^\beta(t) - \gamma \frac{\partial D(\boldsymbol{\mu}^\alpha(t), \boldsymbol{\mu}^\beta(t))}{\partial \mu_f^\beta}]^+ \quad (16)$$

γ is a stepsize. $D(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta)$ is continuously differentiable since U_f is strictly concave[3]. Thus, it follows that

$$\frac{\partial D(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta)}{\partial \mu_l^\alpha} = c_l - \sum_{f \in \mathcal{F}(l)} x_f(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) \quad (17)$$

$$\frac{\partial D(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta)}{\partial \mu_f^\beta} = x_{f^p}(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) - x_f(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) \quad (18)$$

where f^p is the parent flow of f.

Substituting Eq. (17) into (15), (18) into (16), we have

$$\mu_l^\alpha(t+1) \quad (19)$$
$$= [\mu_l^\alpha(t) + \gamma(\sum_{f \in \mathcal{F}(l)} x_f(\boldsymbol{\mu}^\alpha(t), \boldsymbol{\mu}^\beta(t)) - c_l)]^+$$
$$\mu_f^\beta(t+1) \quad (20)$$
$$= [\mu_f^\beta(t) + \gamma(x_f(\boldsymbol{\mu}^\alpha(t), \boldsymbol{\mu}^\beta(t)) - x_{f^p}(\boldsymbol{\mu}^\alpha(t), \boldsymbol{\mu}^\beta(t)))]^+$$

Eq. (19) reflects the law of supply and demand. If the demand for bandwidth at link l exceeds its supply c_l, the network constraint is violated. Thus, the link price μ_l^α is raised. Otherwise, μ_l^α is reduced. Similarly, in Eq. (20), if f demands a flow rate higher than its parent flow f^p, the data constraint is violated. Thus, the relay price μ_f^β is raised. Otherwise, μ_f^β is reduced[1].

Also at time t, when f receives the updated prices $\boldsymbol{\mu}^\alpha(t)$ and $\boldsymbol{\mu}^\beta(t)$, $\lambda_f^\alpha(t)$ and $\lambda_f^\beta(t)$ can be acquired by substituting $\boldsymbol{\mu}^\alpha(t)$ and $\boldsymbol{\mu}^\beta(t)$ into Eq. (9) and (10). Then f can adjust the flow rate x_f by solving Eq. (14).

We summarize our algorithm in Tab. 3, where link l and flow f are deemed as entities capable of computing and communicating[2].

Theorem 1: Define $Y(f) = \sum_l A_{lf} + \sum_{f'} B_{f'f}$, and $\bar{Y} = \max_{f \in \mathcal{F}} Y(f)$; $U(l) = \sum_{f \in \mathcal{F}} A_{lf}$ and $\bar{U} = \max_{l \in \mathcal{L}} U(l)$; $V(f') = \sum_{f \in \mathcal{F}} B_{f'f}$ and $\bar{V} = \max_{f' \in \mathcal{F}} V(f')$; $\bar{Z} = \max\{\bar{U}, \bar{V}\}$; $\bar{\kappa} = \max_{f \in \mathcal{F}} \kappa_f$. Suppose assumptions **A1** and **A2** hold, and the stepsize $0 < \gamma < 2/\bar{\kappa}\bar{Y}\bar{Z}$, then starting from any initial rates $m_f \le x_f(0) \le M_f$, and prices $\boldsymbol{\mu}^\alpha(0) \ge 0$ and $\boldsymbol{\mu}^\beta(0) \ge 0$, every accumulation point $(\boldsymbol{x}^*, \boldsymbol{\mu}^{\alpha*}, \boldsymbol{\mu}^{\beta*})$ of the sequence $(\boldsymbol{x}(t), \boldsymbol{\mu}^\alpha(t), \boldsymbol{\mu}^\beta(t))$ generated by the algorithm in Tab. 3 is primal-dual optimal.

We present the proof in our technical report[16].

[1]Eq. (18) and (20) do not apply when f has no parent flow (flow 1 in Fig. 1). In this case, μ_f^β will always be 0. For the same reason, the **Relay Price Update** part in Tab. 3 is only for those flows which have a parent flow

[2]As these assumptions do not hold in practice, Sec. 4 will discuss the implementation issues of our algorithm.

Link Price Update (by link l): At times $t = 1, 2, \ldots$
1 Receive rates $x_f(t)$ from all flows $f \in \mathcal{F}(l)$
2 Update price
$\mu_l^\alpha(t+1) = [\mu_l^\alpha(t) + \gamma(\sum_{f \in \mathcal{F}(l)} x_f(t) - c_l)]^+$
3 Send $\mu_l^\alpha(t+1)$ to all flows $f \in \mathcal{F}(l)$

Relay Price Update (by flow f): At times $t = 1, 2, \ldots$
1 Receive rate $x_{f^p}(t)$ from its parent flow f^p
2 Update price
$\mu_f^\beta(t+1) = [\mu_f^\beta(t) + \gamma(x_f(t) - x_{f^p}(t))]^+$
3 Send $\mu_f^\beta(t+1)$ to f^p

Stream Rate Adaptation (by flow f): At times $t = 1, 2, \ldots$
1 Receive link prices $\mu_l^\alpha(t)$ from all links $l \in \mathcal{L}(f)$
2 Receive relay prices $\mu_{f'}^\beta(t)$ from all children flows $\{f' \mid f \to f'\}$
3 Calculate
$\lambda_f^\alpha(t) = \sum_{l \in \mathcal{L}(f)} \mu_l^\alpha(t)$
$\lambda_f^\beta(t) = \mu_f^\beta(t) - \sum_{f \to f'} \mu_{f'}^\beta(t)$
4 Adjust rate
$x_f(t+1) = [U_f^{'-1}(\lambda_f^\alpha(t) + \lambda_f^\beta(t))]_{m_f}^{M_f}$
5 Sends $x_f(t+1)$ to all links $l \in \mathcal{L}(f)$ and all children flows $\{f' \mid f \to f'\}$

Table 3. Algorithm

3.4 Example

We use the example in Fig. 1 to illustrate the algorithm. We set $U_f(x_f) = \ln(x_f)$ for each $f \in \mathcal{F}$. The range of U_f is $I_f = [1, \infty)$. The resulting optimal rates are $x_1^* = 2$, $x_2^* = 4$, $x_3^* = 4$, $x_4^* = 2$, $x_5^* = 2$. The aggregate utility is $\sum_{f \in \mathcal{F}} U_f(x_f^*) = 4.852$.

One might wonder if the same result can be obtained if we first acquire the optimal rates by treating each f as independent unicast flow, then enforce the data constraint. We now verify this conjecture. In the first step, we temporarily remove constraint (6) in problem $\mathbf{P}$. Consequently, the relay price vector $\boldsymbol{\mu}^\beta$ is removed from the Lagrangian form (9). $\boldsymbol{\lambda}^\beta$ is also removed. In fact, the problem falls back to the unicast flow rate allocation, whose details can be found at [10]. Reflected in the algorithm, the rate adaptation function is modified as

$$x_f(t+1) = [U_f^{'-1}(\lambda_f^\alpha(t))]_{m_f}^{M_f}$$

Finally, we get a different set of optimal rates: $x_1^* = 3$, $x_2^* = 3$, $x_3^* = 5$, $x_4^* = 2$, $x_5^* = 2$. In the second step, we reapply constraint (6) to this set of rates. As a result, x_3^* is changed to 3, in accordance with its parent flow rate x_2^*. Now the aggregate utility is $\sum_{f \in \mathcal{F}} U_f(x_f^*) = 4.682$, which is suboptimal to the original result.

The reason lies at the link from Host 1 to R_1 (Fig. 1 (b)). Flows 1 and 2 share this bottleneck link. In the alternative approach, these two flows equally share the bottleneck bandwidth. In fact, flow 2 has a subtree of children flows, while flow 1 has no children at all. Apparently, flow 2 should be assigned more bandwidth, as it can get more relay benefit to increase the utility of its children, hence the aggregate utility. This example confirms our argument that both network constraint and data constraint have to be simultaneously addressed, which is a unique property of the optimal resource allocation problem in overlay multicast.

4 Protocol Design and Implementation

The algorithm presented in Sec. 3.3 treat each flow f and link l as entities capable of computing and communicating. In practice, we propose to let end hosts delegate the tasks of f and l. This idea has not been explored by existing works[10][8], which assume that the network link (actually the router connected to it) is capable of measuring flow rates, calculating link price, and hence updating price signal to the end host, none of which exists in the current Internet.

4.1 Assumptions

First, we assume that a flow f's rate is controlled and adjusted by the end host, denoted as the *flow owner*, O_f. If the flow rate adaptation is *receiver-based*, O_f is the receiver of f. Otherwise, in *sender-based* rate adaptation, O_f is the sender of f.

Second, we assume that the underlying route of a flow path can be found by network path finding tools such as *traceroute*. This enables an end host to have explicit knowledge of what physical links are passed through its unicast path. Considering the fact that most routes in Internet today is relatively stable, the update overhead is small.

Our final assumption is that the available bandwidth of each physical link can be measured by tools such as *pathrate*[2], in an end-to-end manner.

4.2 Protocols

We design two protocols, *Link Delegation* and *Aggregated Route Pricing*. They are exclusive and each of them works independently. We will compare their performances in Sec. 5.

4.2.1 Protocol A: Link Delegation

In this protocol, each physical link l is assigned to an end host, denoted as D_l, which delegates the task of l. D_l measures the available bandwidth of l. For all flows sharing l, we collect their owners into a set $\mathcal{C}(l)$ as below.

Notation	Definition
O_f ($f \in \mathcal{F}$)	Flow Owner of f
D_l ($l \in \mathcal{L}$)	Link Delegate of l
$\mathcal{C}(l)$ ($l \in \mathcal{L}$)	Set of Hosts whose Flows Go Through l
$\mathcal{N}(f)$ ($f \in \mathcal{F}$)	Set of Flow Owners f needs to Calculate λ_f^β
$\mathcal{Q}(f)$ ($f \in \mathcal{F}$)	Set of Flows Sharing a Link with f
$\mathbb{P}_h$ ($h \in \mathcal{H}$)	The Set Storing **LPU** messages
$\mathbb{R}_h$ ($h \in \mathcal{H}$)	The Set Storing **FRR** messages
$\mathbb{L}_h$ ($h \in \mathcal{H}$)	$\{l \mid D_l = h\}$
$\mathbb{F}_h$ ($h \in \mathcal{H}$)	$\{f \mid O_f = h\}$

Table 4. Notations in Sec. 4

$$\mathcal{C}(l) = \{O_f \mid f \in \mathcal{F}(l)\} \tag{21}$$

All hosts in $\mathcal{C}(l)$ periodically report their streaming rates to D_l. According to these rates, D_l updates the link price based on Eq. (19), and sends it back to the reporting hosts. For a flow f, upon receiving the price feedbacks from all hosts delegating links on its path, i.e., $\{D_l \mid l \in \mathcal{L}(f)\}$, its owner O_f is able to calculate the network price λ_f^α based on Eq. (9).

To calculate the data price λ_f^β, f must know the relay prices of itself and its children flows $\{f' \mid f \rightarrow f'\}$ (Eq. (10)). By Eq. (20), the relay price of a flow f is calculated based on its own rate x_p and its parent flow rate x_{f^p}. It means that λ_f^β can be solved once the rate of its parent flow f^p and all children flow $\{f' \mid f \rightarrow f'\}$ are available to f. From the flow owner's point of view, O_f can independently calculate λ_f^β if it receives the stream rate report from all hosts in the set $\mathcal{N}(f)$.

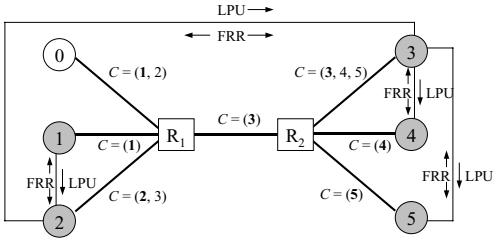
$$\mathcal{N}(f) = O_{f^p} \cup \{O_{f'} \mid f \rightarrow f'\} \tag{22}$$

Consider an end host h, which is both the flow owner O_f of some flow f, and the link delegate D_l of some link l. Then it is possible that $\mathcal{N}(f) \cap \mathcal{C}(l) \neq \phi$. Therefore, messaging overhead can be saved if we maximize this intersection set by choosing D_l according to the following rules.

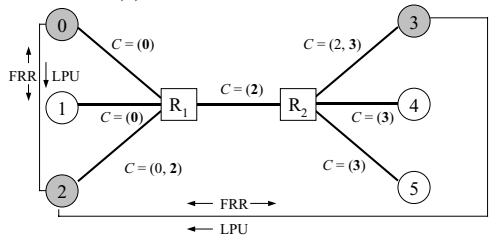
1. It must satisfy that $D_l \in \mathcal{C}(l)$.
2. If l is an access link connecting some end host h, then it follows that $D_l = h$, if the first rule is not violated.

We use the same example in Fig. 1 to illustrate the above rules. In Fig. 2, a host is grayed if it acts as the delegate of some links. Each link l is marked with $\mathcal{C}(l)$, the set of all hosts sharing l. Inside $\mathcal{C}(l)$, the bolded one is the selected

link delegate. In Fig. 2 (a), the link from R_1 to R_2 is delegated by host 3 (based on Rule 1). In this way, it saves to send message to itself. Host 3 also delegates the access link from itself to R_2 (based on Rule 2), as this link is shared by all its children flows (recall the second assumption in Sec. 4.1). Therefore, the owners of the children flows, hosts 4 and 5, belong to both $\mathcal{N}(f)$ and $\mathcal{C}(l)$. As a result, they only need to report their stream rates to host 3 once.



(a) Receiver-based Protocol



(b) Sender-based Protocol

Figure 2. Protocol A: Link Delegation

We present the protocol in Tab. 6. The message formats are listed in Tab. 5. There are two types of messages: *Flow Rate Report* (**FRR**) and *Link Price Update* (**LPU**). Each end host h maintains the following sets.

1. $\mathbb{P}_h$ caches all received **LPU** messages.
2. $\mathbb{R}_h$ caches all received **FRR** messages.
3. $\mathbb{L}_h$ stores all links that h delegates.
4. $\mathbb{F}_h$ stores all flows that h owns.

Link Price Update (LPU)		**Flow Rate Report (FRR)**	
$\langle l \rangle$	link	$\langle f \rangle$	flow
$\langle \mu_l^\alpha \rangle$	link price	$\langle x_f \rangle$	flow rate
$\langle D_l \rangle$	link delegate	$\langle O_f \rangle$	flow owner

Table 5. Message Formats

4.2.2 Protocol B: Aggregated Route Pricing

In this protocol, we let each flow f individually calculates its own network price. The flow owner O_f calculates the price of each link on its path ($\{p_l \mid l \in \mathcal{L}(f)\}$), then adds them up to acquire the network price, finally adjusts the streaming rate. Thus, **LPU** messages are not required. To achieve this, O_f needs to measure the available bandwidth of all links on its path. Besides, it has to receive the **FRR** message from every other flow, which shares at least one link with f. These flows are collected into a set $\mathcal{Q}(f)$, defined as follows.

$$\mathcal{Q}(f) = \{f' \mid \mathcal{L}(f') \cup \mathcal{L}(f) \neq \phi\} \tag{23}$$

End Host h

Initialization
1 $\mathbb{L}_h \leftarrow \{l \mid D_l = h\}$
2 $\mathbb{F}_h \leftarrow \{f \mid O_f = h\}$
3 $\mathbb{P}_h \leftarrow \phi$
4 $\mathbb{R}_h \leftarrow \phi$

On Receiving **FRR** Message
1 Read $\langle f \rangle$, $\langle x_f \rangle$ and $\langle O_f \rangle$ fields of the message
2 $\mathbb{R}_h \leftarrow \mathbb{R}_h \cup f$
3 **for** $\forall l \in \mathbb{L}_h$ **that** $\mathcal{F}(l) \subseteq \mathbb{R}_h$
4 $\mu_l^\alpha \leftarrow [\mu_l^\alpha + \gamma(\sum_{f \in \mathcal{F}(l)} x_f - c_l)]^+$
5 Send **LPU** message to each $\{O_f \mid f \in \mathcal{F}(l)\}$, setting $\langle l \rangle \leftarrow l$, $\langle \mu_l^\alpha \rangle \leftarrow \mu_l^\alpha$ and $\langle D_l \rangle \leftarrow h$
6 $\mathbb{L}_h \leftarrow \mathbb{L}_h - l$
7 **Goto** Stream Rate Update

On Receiving **LPU** Message
1 Read $\langle l \rangle$, $\langle \mu_l^\alpha \rangle$ and $\langle D_l \rangle$ fields of the message
2 $\mathbb{P}_h \leftarrow \mathbb{P}_h \cup l$
3 **Goto** Stream Rate Update

Stream Rate Update
1 **for** $\forall f \in \mathbb{F}_h$ **that** $\mathcal{L}(f) \subseteq \mathbb{P}_h$ **and** $f^p \in \mathbb{R}_h$ **and** $\{f' \mid f \rightarrow f'\} \subseteq \mathbb{R}_h$
2 $\lambda_f^\alpha \leftarrow \sum_{l \in \mathcal{L}(f)} \mu_l^\alpha$
3 $\mu_f^\beta \leftarrow [\mu_f^\beta + \gamma(x_f - x_{f^p})]^+$
4 **for each** $\{f' \mid f \rightarrow f'\}$
5 $\mu_{f'}^\beta \leftarrow [\mu_{f'}^\beta + \gamma(x_{f'} - x_f)]^+$
6 $\lambda_f^\beta \leftarrow \mu_f^\beta - \sum_{f \rightarrow f'} \mu_{f'}^\beta$
7 $x_f \leftarrow [U_f^{'-1}(\lambda_f^\alpha + \lambda_f^\beta)]_{m_f}^{M_f}$
8 Send **FRR** message to each $\{D_l \mid l \in \mathcal{L}(f)\}$, setting $\langle f \rangle \leftarrow f$, $\langle x_f \rangle \leftarrow x_f$, $\langle O_f \rangle \leftarrow h$
9 $\mathbb{F}_h \leftarrow \mathbb{F}_h - f$
10 **if** $\mathbb{F}_h = \phi$ **and** $\mathbb{F}_h = \phi$
11 **Goto** Initialization

Table 6. Protocol A: Link Delegation

We find out that $\mathcal{N}(f)$ is a subset of all hosts that owns $\mathcal{Q}(f)$. By the single access link assumption in Sec. 4.1,

a flow f shares the access link with all its children at the receiver side, and shares the access link with its parent at the sender side. Therefore, f's data price λ_f^β can be calculated once all messages from flows in $\mathcal{Q}(f)$ are received. We present the protocol in Tab. 7.

End Host h

Initialization

1 $\mathbb{F}_h \leftarrow \{f \mid O_f = h\}$
2 $\mathbb{R}_h \leftarrow \phi$

On Receiving **FRR** Message

1 Read $\langle f \rangle$, $\langle x_f \rangle$ and $\langle O_f \rangle$ fields of the message
2 $\mathbb{R}_h \leftarrow \mathbb{R}_h \cup f$
3 **for** $\forall f \in \mathbb{F}_h$ **that** $\mathcal{Q}(f) \subseteq \mathbb{R}_h$
4 **for each** $l \in \mathcal{L}(f)$
5 $\mu_l^\alpha \leftarrow [\mu_l^\alpha + \gamma(\sum_{f \in \mathcal{F}(l)} x_f - c_l)]^+$
6 $\lambda_f^\alpha \leftarrow \sum_{l \in \mathcal{L}(f)} \mu_l^\alpha$
7 $\mu_f^\beta \leftarrow [\mu_f^\beta + \gamma(x_f - x_{f^p})]^+$
8 **for each** $\{f' \mid f \rightarrow f'\}$
9 $\mu_{f'}^\beta \leftarrow [\mu_{f'}^\beta + \gamma(x_{f'} - x_f)]^+$
10 $\lambda_f^\beta \leftarrow \mu_f^\beta - \sum_{f \rightarrow f'} \mu_{f'}^\beta$
11 $x_f \leftarrow [U_f^{'-1}(\lambda_f^\alpha + \lambda_f^\beta)]_{m_f}^{M_f}$
12 Send **FRR** message to each $\{O_f \mid l \in \mathcal{Q}(f)\}$, setting $\langle f \rangle \leftarrow f$, $\langle x_f \rangle \leftarrow x_f$, $\langle O_f \rangle \leftarrow h$
13 $\mathbb{F}_h \leftarrow \mathbb{F}_h - f$
14 **if** $\mathbb{F}_h = \phi$
15 **Goto** Initialization

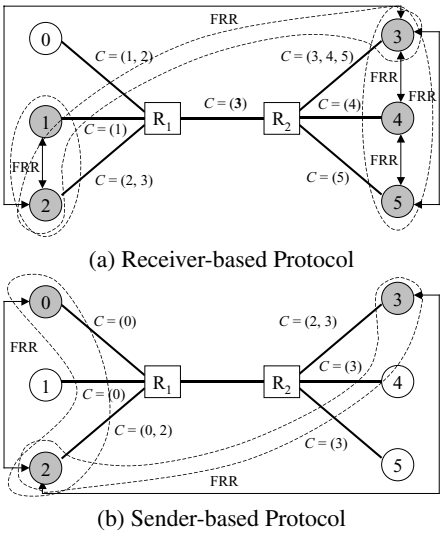
Table 7. Protocol B: Aggregated Route Pricing

Again, we use the example in Fig. 1 to illustrate the protocol. We start with the receiver-based protocol. In Fig. 3 (a), all receivers are grayed. From the definition of $\mathcal{Q}(f)$, we can easily derive that if $f' \in \mathcal{Q}(f)$, then $f \in \mathcal{Q}(f')$. Thus, various "cliques" are formed. Hosts within a clique exchange **FRR** messages to each other. In Fig. 3 (a), we find three cliques, $\{1, 2\}$, $\{2, 3\}$ and $\{3, 4, 5\}$. Above illustrations also apply for the sender-based protocol (Fig. 3 (b)), which will not be elaborated.

5 Simulation Results

5.1 Experimental Setup

We use the Boston BRITE[1] topology generator to setup our experimental network. We choose the hierarchical topology model. We first generate an AS-level topology consisting of 10 nodes. Each node in the AS-level topology generates a router-level topology of 100 nodes. Therefore, the size of our experimental network is 1000 nodes. Each



(a) Receiver-based Protocol

(b) Sender-based Protocol

Figure 3. Protocol B: Aggregated Route Pricing

overlay node is an end host attached to a single router. The bandwidths of all links are uniformly distributed between 10 and 100 Mbps. The average propagation delay of each individual link is 1.20 ms. A single overlay multicast session runs on our experimental network. The multicast tree is constructed as follows. Each new host h attaches itself to one of the existing multicast members, which is closest to h in terms of end-to-end latency, and whose degree in the multicast tree is less than k. In our experiment, $k = 4$. We present a subset of our results in this paper. Others can be found in our technical report[16].

5.2 Flow Rate Convergence

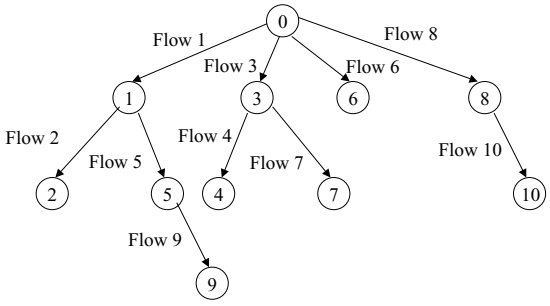


Figure 4. Experimental Overlay Multicast Tree

We first test the performance of our algorithm at converging to the optimal flow rate. We setup an overlay multicast session of 10 members. The multicast tree is shown

Rate($Mbps$)	x_1^*	x_2^*	x_3^*	x_4^*	x_5^*	x_6^*	x_7^*	x_8^*	x_9^*	x_{10}^*
Overlay	13.75	13.75	9.24	9.24	13.75	12.58	9.24	25.15	13.75	25.15
Unicast	12.67	14.41 (12.67)	6.91	10.41 (6.91)	27.56 (12.67)	20.57	10.41 (6.91)	20.57	35.00 (12.67)	26.91 (20.57)

Table 8. Optimal Rate Comparison of Overlay-based and Unicast-based Resource Allocation Schemes

in Fig. 4. Host 0 is the server. Initially, host 1 joins the session. In every minute thereafter, a new member joins. Each member updates is flow rate every 0.1 second. The step size (γ in Eq. (15) and (16)) is 0.0005. The utility function of every flow f is $U_f(x_f) = \ln(x_f)$. The minimal rate (m_f in Eq. (14)) is 1 Mbps. The maximal rate (M_f in Eq. (14)) is 35 Mbps.

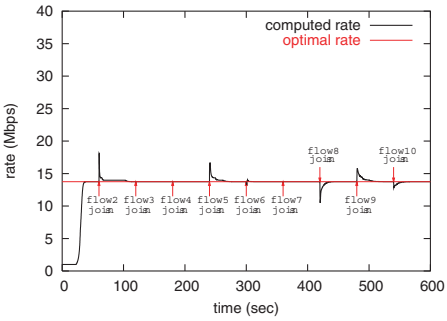


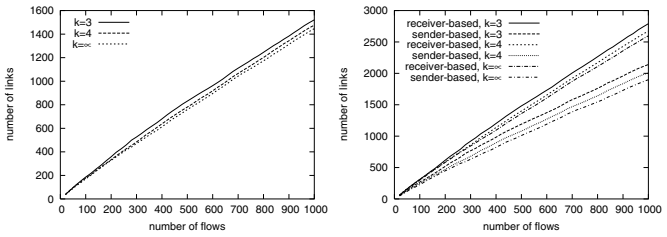
Figure 5. Convergence of Flow Rates (Flow 1)

We choose to show the rate adaptation procedure of Flow 1 in Fig. 5. It is shown that the computed rates track close to the optimal rates. They are disturbed when new members join the multicast session, but quickly converge back to the optimal rates within normally 30 seconds (300 iterations).

The final optimal rates of all flows are shown in Tab. 8. The aggregate utility is $\sum_{f=1}^{10} U_f(x_f^*) = 26.14$. We also compute the optimal rates using the unicast-based resource allocation mechanism reported in [10], without considering the data constraint. These rates are then adjusted so that they are no higher than their parent flow rates (as listed in parentheses). The aggregate utility of all adjusted flow rates is 25.03, which is suboptimal to the result of overlay-based mechanism.

5.3 Link Measurement Overhead

We now proceed to evaluate the performance of our protocols. One of their tasks is to periodically measure the available bandwidths of network links, through which the overlay flows travel. The network price of a flow can be determined only when the available bandwidths of all links along its path is known. Now we measure the overhead of this task.

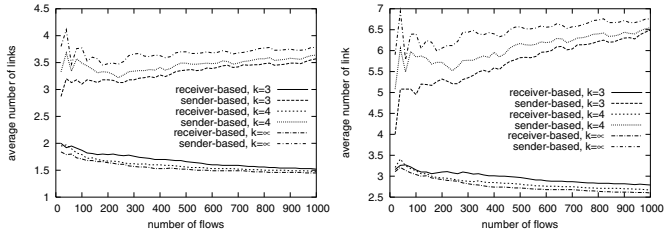


(a) Link Delegation (b) Aggregated Route Pricing

Figure 6. Total Link Measurement Overhead

Fig. 6 shows the overall link measurement overhead, i.e., the total number of measurement operations that have to be performed by the entire multicast session. For *Link Delegation* protocol (Fig. 6 (a)), its curves actually records number of links the multicast session contains, since each link is uniquely assigned to a single end host as its delegate, which is entitled to carry out the measurement operation. However, in *Aggregated Route Pricing* protocol, each flow independently measures the available bandwidth of all links along its path, regardless of whether some link has already been measured by other flows sharing it. Therefore, in Fig. 6 (b), the total number of operations exceeds the total number of links, suggesting that some links have been repeatedly measured. In Fig. 6 (b), it is also observed that the sender-based protocol is more efficient than the receiver-based protocol. The reason is that there are always fewer senders than receivers in a multicast session. While each flow is owned by a unique receiver, a sender normally owns multiple flows. Consider a link shared by two flows originated from the same sender , e.g., the link from Host 1 to R_1 in Fig. 3. In case of receiver-based protocol, this link has to be measured twice by each receiver. In case of the sender-based protocol, it only needs to be measured once by the common sender of the two flows.

Fig. 7 shows the average link measurement overhead per host. For both *Link Delegation* and *Aggregated Route Pricing*, their receiver-based protocols exhibits great scalability. The average number of link measurement operations per receiver slightly decreases as the multicast session expands. However, for the sender-based protocols, the same overhead is almost doubled, although the sender-based approach is more efficient than the receiver-based approach in terms of overall measurement overhead. The reason is, since there



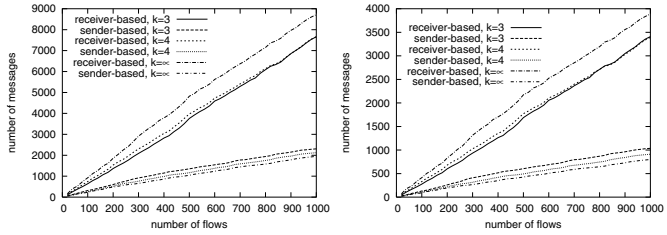
(a) Link Delegation (b) Aggregated Route Pricing

Figure 7. Average Link Measurement Overhead per Host

are fewer senders than receivers, and only senders are entitled to participate the link measurement, the average load on each sender is aggravated, compared to the receiver-based protocols. Furthermore, relaxing the degree constraint also has a negative effect here: increasing k results in even fewer number of senders. Thus, each sender can be further overloaded. This explains why in both Fig. 7 (a) and (b), the top curves belong to the case of sender-based protocol and no degree constraint.

5.4 Messaging Overhead

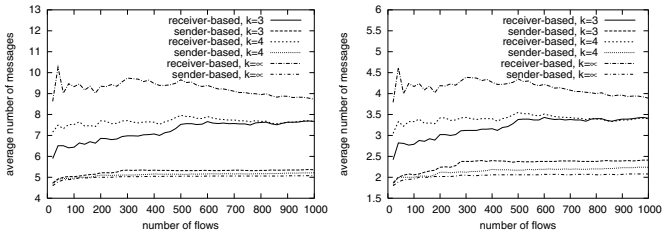
Another task of our protocols is exchanging **LPU** and **FRR** messages among end hosts to facilitate the link price update and flow rate adaptation. We now evaluate the messaging overhead.



(a) Link Delegation (b) Aggregated Route Pricing

Figure 8. Total Messaging Overhead

Fig. 8 shows the overall messaging overhead, the total number of messages sent out in one iteration of flow rate adaptation. We have the following observations. First, the *Aggregated Route Pricing* is more efficient than *Link Delegation*, since the former approach eliminates the need for **LPU** messages. Second, the sender-based approach is more efficient than the receiver-based approach. This observation can be illustrated by the same example in Sec. 5.3. Consider a link l shared by two flows, which have the same sender. In receiver-based approach, the receivers of these flows have to exchange **FRR** messages to each other, in order to calculate the price of l. In sender-based approach, the sender owns both flows on l, which enables it to calculate the price of l independently without any message exchange.



(a) Link Delegation (b) Aggregated Route Pricing

Figure 9. Average Messaging Overhead per Host

In Fig. 9, we have the same observations as in Fig. 8. All protocols are scalable, since the average messaging overhead per host remains stable as the size of the multicast session grows.

In both Fig. 8 and 9, we notice that relaxing the degree constraint has opposite impacts on sender-based protocols and receiver-based protocols. For the sender-based approach, increasing k results in less messaging overhead. But for the receiver-based approach, it does the opposite. To clarify this seemingly contradict observation, we first explore the double-sided effect of choosing degree constraint. On one hand, increasing k helps reduce the number of links in a multicast session, which in turn helps reduce the total number of messages for link price calculation. On the other hand, it increases the flow concurrency on network links, especially those closed to the sender side, which makes each link to cost more messages. For receiver-based approach, the negative effect gains the leading edge, since for all flows sharing one link, each of them is owned by a different receiver, which introduces more message exchanges around each individual link. However, for sender-based approach, it is often the case that many of these flows are owned by a common sender, which does not need any message exchange at all. In this case, the positive effect wins over the negative one.

We conclude the experimental results of Sec. 5.3 and 5.4 as follows. First, *Aggregated Route Pricing* yields fewer messages than *Link Delegation*, as it avoids the need for communication on link price updates. On the flip side though, it introduces more link measurement overhead, since in this approach, a link shared by multiple flows has to be measured repeatedly by their flow owners. Second, the sender-based approach is more efficient than the receiver-based approach on both messaging and link measurement overhead. The fundamental reason is that in receiver-based approach, the flow information are distributed within the

group of receivers, which contains all multicast members. In sender-based approach, the same information are limited within the group of senders, a subset of multicast members. Clearly, a smaller group introduces less communication and control overhead. However, the side effect is that each individual member in this group can be overloaded compared to the receiver-based approach. Finally, the protocol overhead is affected by the way the multicast tree is built. We have explored the double-sided effect of relaxing the degree constraint: increasing k results in fewer number of network links in a multicast session, but increases average flow concurrency on each link. This effect has totally opposite impacts on different protocols.

6 Related Work

The price-based resouarce allocation strategies have been extensively explored in the context of IP unicast and multicast. In [5] and [6], Kelly et al. associate a shadow price with each network link. The prices work as signals to reflect the traffic load, and the end hosts choose a transmission rate to optimize its net benefit, i.e., the difference of its utility and network cost. Low et al.[10] then presents a distributed algorithm based on the dual approach of the same problem. Kar et al. are the first to apply the price-based resource allocation mechanism into multirate multicast. They design a distributed algorithm using subgradient projection and proximal approximation techniques[3].

The fundamental difference of our work to the above ones, as we have argued in Sec. 1, is that our resource allocation scheme incorporates the data constraint, a unique challenge only taking place in the scenario of overlay multicast. Plus, all previous works require the underlying network to be capable of measuring network traffic, calculating and communicating price signals, which is rather unrealistic in the context of overlay network.

In a broader sense, overlay resource allocation should not only include the network resource, which this paper focuses on, but also resources of end hosts within the overlay network, such as CPU and storage. [12] presents a global flow control scheme to manage overlay resources, including bandwidth and buffer space of overlay routers. Opus[9] is an overlay utility service, which provides a unified platform to allocate utility resources, such as end system CPU and storage, among competing applications. Our previous works[15][14] have explored the optimal utilization of end host buffer spaces to facilitate overlay-based multimedia distribution.

7 Conclusions

This paper targets the problem of optimal network resource allocation in overlay multicast. We identify both theoretical and practical challenges from this problem. Theoretically, resource allocation among overlay flows is not only subject to the network capacity constraint, but also the data availability constraint. Practically, our solution has to be purely end-host-based in accordance with the design objective of overlay network. We propose a distributed algorithm, which maximizes the aggregate utility of all multicast members. We then implement our algorithm in a series of end-host-based protocols. Our experiments prove the scalability and efficiency of our solution.

References

[1] A. Medina, A. Lakhina, I. Matta, and J. Byers. Brite: An approach to universal topology generation. In *IEEE MASCOTS*, 2001.

[2] C. Dovrolis, P. Ramanathan, and D. Moore. What do packet dispersion techniques measure? In *IEEE INFOCOM*, 2001.

[3] D. Bertsekas and J. Tsitsiklis. *Parallel and Distributed Computation*. Prentice-Hall, 1989.

[4] E. Amir, S. McCanne, and R. Katz. An active service framework and its application to real-time multimedia transcoding. In *ACM SIGCOMM*, 1998.

[5] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8(1), 1997.

[6] F. Kelly, A. Maulloo and D. Tan. Rate control for communication networks: Shadow prices, proportional fairness and stability. *Journal of Operations Research Society*, 49(3), 1998.

[7] K. Kar, S. Sarkar and L. Tassiulas. A low-overhead rate control algorithms for maximizing aggregate receiver utility for multirate multicast sessions. In *SPIE ITCOM*, 2001.

[8] K. Kar, S. Sarkar and L. Tassiulas. Optimization basd rate control for multirate multicast sessions. In *IEEE INFOCOM*, 2001.

[9] R. Braynard, D. Kostic, A. Rodriguez, J. Chase and A. Vahdat. Opus: An overlay utility service. In *OPENARCH*, 2002.

[10] S. Low and D. Lapsley. Optimization flow control, i: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7(6), 1999.

[11] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven layered multicast. In *ACM SIGCOMM*, 1996.

[12] Y. Amir, B. Awerbuch, C. Danilov and J. Stanton. Global flow control for wide are overlay networks: a cost-benefit approach. In *OPENARCH*, 2002.

[13] Y. Chu, R. Rao, and H. Zhang. A case for end system multicast. In *ACM SIGMETRICS*, 2000.

[14] Y. Cui and K. Nahrstedt. Layered peer-to-peer streaming. In *NOSSDAV*, 2003.

[15] Y. Cui, B. Li and K. Nahrstedt. ostream: Asynchronous streaming multicast in application-layer overlay networks. *to appear in IEEE JSAC special issue on Recent Advances in Service Overlay*, 2003.

[16] Y. Cui, Y. Xue and K. Nahrstedt. Optimal resource allocation in overlay multicast. *Technical Report UIUCDCS-R-2003-2373/UILU-ENG-2003-1760*, 2003.