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

In overlay multicast, each receiver of the multicast group is free to choose its streaming rate subject to various constraints such as network capacity. We model the rate allocation in overlay multicast as a utility-based optimization problem. We associate a utility with each receiver, which is defined as a function of its streaming rate. Our goal is to maximize the aggregate utility of all receivers. We identify two constraints for this problem: network capacity constraint, and data constraint, which is unique in overlay multicast, mainly due to the dual role of end hosts as both receivers and senders. Based on this theoretical formalization, we propose distributed algorithms in synchronous and asynchronous network settings, both of which are proved to converge to the optimal point, where the aggregate utility of all receivers is maximized. We implement our algorithms using an end-host-based protocol. In contrast to traditional resource allocation schemes, which assume the network links to be capable of measuring flow rates, calculating and communicating control signals, our protocol purely 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.

I. Introduction

A. Motivation

Multicast is an important communication paradigm to support many network applications, such as teleconferencing, multimedia distribution, etc. In this paper, we are particularly interested in overlay multicast[1], a special form of multicast, where end hosts self-organize into an overlay network and accomplish multicast by relaying data to each other via unicast. Overlay multicast not only provides a working solution to address the deficiency of infrastructure support, i.e., IP multicast is largely unavailable in the Internet, but also marks a paradigm shift, which radically changes the way network applications can be built. In IP multicast, the network is mainly composed of routers, whose task is no more than forwarding packets. In contrast, each overlay node is an intelligent one that can carry out more sophisticated operations, and contribute various resources such as their CPU power, storage space, and access bandwidth.

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.

A good example to illustrate such a paradigm shift is to compare these two types of solutions at supporting multi-rate multicast, where heterogeneous receivers in the same group can retrieve service at different rates. In IP-multicast-based solutions, this is mainly achieved by layered streaming[2][3]. Here, 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[4], all end-to-end stream adaptation techniques (frame dropping, transcoding[5], etc.) can be applied, since the data relay happens on each end host, and each data link actually represents a unicast path. In this way, the receiver is allowed to choose its streaming rate on a continuous range.

B. Challenges

If each receiver in a multicast group is free to choose its streaming rate over a certain range, does there exist an optimal rate allocation which maximizes the utilization of network resource, meanwhile maintaining certain fairness among all receivers? If so, how to achieve this goal? Is it doable in a distributed manner, where each end host makes its own rate adaptation decision without any central authority involved? This paper tries to answer these questions.

An optimal rate allocation 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[6][7]. Utility-based resource allocation has been explored in the rate control of unicast[8][7] and IP multicast[9][10]. 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 completely 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. III-D to justify our argument.

Practically, existing solutions[8][10] 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.

C. Contributions

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[8], 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". It is proved that the rate allocation converges to the optimal point, at which the aggregate utility of all receivers is maximized. We then extend our algorithm to the asynchronous setting, i.e., the flow rate and price update do not need to be synchronized, and prove that all properties of the original algorithm still hold.

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. It purely relies on the coordination of end hosts to calculate and exchange network/data price signals, and adjust the flow rate. In contrast to past solutions[8][10], 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. II introduces the network model. Sec. III presents the problem formulation and proposes a distributed algorithm. Sec. IV extends the algorithm into the asynchronous setting. Sec. V discusses the protocol design and implementation in overlay network environment. Finally, we show experimental results in Sec. VI, discuss the related work in Sec. VII, and conclude in Sec. VIII.

II. 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 [1], denoted as $\mathcal{F} = \{1, 2, \ldots, F\}$. Each flow $f \in \mathcal{F}$ has a rate x_f. We collect them into 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 is determined by unicast routing in the physical 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.

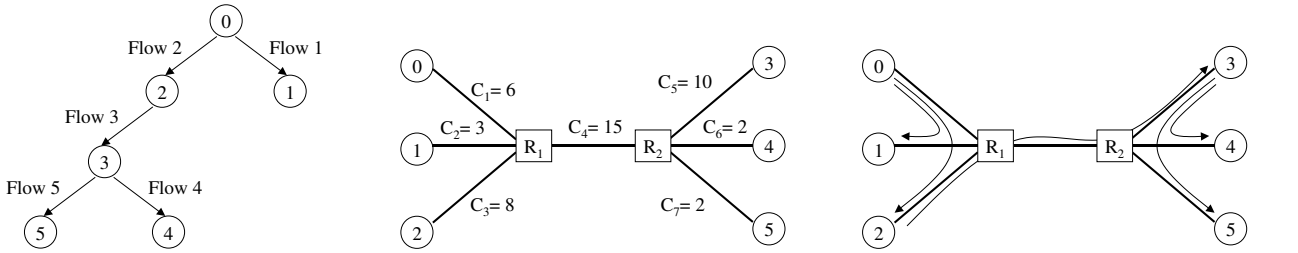
$$B_{f'f} = \begin{cases} -1 & \text{if } f \rightarrow f' \\ 1 & \text{if } f' = f \text{ and } f \text{ has a parent flow} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$\boldsymbol{B}$ specifies the relaying relationship and data dependency in overlay multicast. It is determined by the overlay multicast tree[1]. 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. II. In the example by Fig. 1, there are 5 overlay multicast flows

[1]It is obvious that in overlay multicast, $F = H - 1$



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

Fig. 1. Sample Illustrating Overlay Multicast

($F = 5$). The physical network consists of 7 links ($L = 7$). Hence, Inequality (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 I

NOTATIONS IN SEC. II

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

III. Optimal resource allocation

A. 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.

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

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, which can be solved by Lagrangian method. Let us consider the Lagrangian form of this optimization problem:

$$L(\boldsymbol{x}, \boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) = \sum_{f\in\mathcal{F}} U_f(x_f) - \boldsymbol{\mu}^\alpha(\boldsymbol{A}\cdot\boldsymbol{x} - \boldsymbol{c}) - \boldsymbol{\mu}^\beta(\boldsymbol{B}\cdot\boldsymbol{x}) \quad (8)$$

$\boldsymbol{\mu}^\alpha = (\mu_l^\alpha, l \in \mathcal{L})$ and $\boldsymbol{\mu}^\beta = (\mu_f^\beta, f \in \mathcal{F})$ are vectors of Lagrangian multipliers. Eq. (8) can be further derived as follows.

$$\begin{aligned} L(\boldsymbol{x}, \boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) &= \sum_{f\in\mathcal{F}} U_f(x_f) - \sum_{l\in\mathcal{L}} \mu_l^\alpha (\sum_{f\in\mathcal{F}} A_{lf}x_f - c_l) - \sum_{f'\in\mathcal{F}} \mu_{f'}^\beta (\sum_{f\in\mathcal{F}} B_{f'f}x_f) \quad (9)\\ &= \sum_{f\in\mathcal{F}} U_f(x_f) - \sum_{f\in\mathcal{F}} x_f \sum_{l\in\mathcal{L}} \mu_l^\alpha A_{lf} - \sum_{f\in\mathcal{F}} x_f \sum_{f'\in\mathcal{F}} \mu_{f'}^\beta B_{f'f} + \sum_{l\in\mathcal{F}} \mu_l^\alpha c_l \end{aligned}$$

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.

$$\lambda_f^\alpha = \sum_{l \in \mathcal{L}} \mu_l^\alpha A_{lf} = \sum_{l \in \mathcal{L}(f)} \mu_l^\alpha \tag{10}$$

$$\lambda_f^\beta = \sum_{f' \in \mathcal{F}} \mu_{f'}^\beta B_{f'f} = \mu_f^\beta - \sum_{f \to f'} \mu_{f'}^\beta \tag{11}$$

Now Eq. (9) becomes

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

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. (10)) is the summation 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. (11)) 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).

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 II
NOTATIONS IN SEC. III

B. 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.

$$\mathbf{D}: \min_{\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta \geq \mathbf{0}} D(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) \tag{12}$$

where

$$\begin{aligned} & D(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) \\ = \; & \max_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) \\ = \; & \max_{\boldsymbol{x}} \sum_{f \in \mathcal{F}} \underbrace{(U_f(x_f) - (\lambda_f^\alpha + \lambda_f^\beta) x_f)}_{\Phi(x_f)} + \sum_{l \in \mathcal{L}} \mu_l^\alpha c_l \end{aligned} \tag{13}$$

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 [11]). Now we have

$$D(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) = \sum_{f \in \mathcal{F}} \max_{x_f \in I_f} \{\Phi(x_f)\} + \sum_{l \in \mathcal{L}} \mu_l^\alpha c_l \tag{14}$$

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

$$x_f(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) = \arg\max_{x_f \in I_f} \{\Phi(x_f)\} = [U_f'^{-1}(\lambda_f^\alpha + \lambda_f^\beta)]_{m_f}^{M_f} \tag{15}$$

By Assumption **A1**, $I_f = [m_f, M_f]$ is the feasible region of $U_f(x_f)$. Therefore, x_f must be no greater than M_f and no less than m_f. Since U_f is concave and the constraints (5) and (6) are linear, there is no duality gap (Proposition 5.2.1 in [12]). Also, the dual optimal prices for Lagrangian multipliers ($\boldsymbol{\mu}^\alpha$ and $\boldsymbol{\mu}^\beta$) exist (Proposition 5.1.4 in [12]), denoted as $\boldsymbol{\mu}^{\alpha *}$ and $\boldsymbol{\mu}^{\beta *}$. If $\boldsymbol{\mu}^{\alpha *} \geq 0$ and $\boldsymbol{\mu}^{\beta *} \geq 0$ are dual optimal, then $x_f(\boldsymbol{\mu}^{\alpha *}, \boldsymbol{\mu}^{\beta *})$ is also primal optimal, given that x_f is primal feasible (Proposition 5.1.5 in [12]).

Now we can claim that once the optimal prices $\boldsymbol{\mu}^{\alpha *}$ and $\boldsymbol{\mu}^{\beta *}$ are available, the optimal rate x_f^* can be achieved by solving Eq. (15). The role of $\boldsymbol{\mu}^\alpha$ and $\boldsymbol{\mu}^\beta$ is two-fold. First, they serve as the pricing signal for

a flow f to adjust its rate x_f. Second, they decouple the primal problem $\mathbf{P}$ (global utility optimization) into invidual rate optimization by each flow $f \in \mathcal{F}$.

C. Algorithm

We solve the dual problem $\mathbf{D}$ using gradient projection method[11]. In this method, $\boldsymbol{\mu}^\alpha$ and $\boldsymbol{\mu}^\beta$ are adjusted in opposite direction to the gradient $\nabla D(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta)$:

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

$$\mu_f^\beta(t+1) = [\mu_f^\beta(t) - \gamma \frac{\partial D(\boldsymbol{\mu}^\alpha(t), \boldsymbol{\mu}^\beta(t))}{\partial \mu_f^\beta}]^+ \tag{17}$$

γ is a stepsize. Substituting Eq. (15) into (14), we have

$$D(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) = \sum_{f \in \mathcal{F}} (U_f(x_f(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta)) - (\lambda_f^\alpha + \lambda_f^\beta) x_f(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta)) + \sum_{l \in \mathcal{L}} \mu_l^\alpha c_l \tag{18}$$

$D(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta)$ is continuously differentiable since U_f is strictly concave[11]. Thus, it follows that

$$\frac{\partial D(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta)}{\partial \mu_l^\alpha} = c_l - \sum_{f \in \mathcal{F}(l)} x_f(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) \tag{19}$$

$$\frac{\partial D(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta)}{\partial \mu_f^\beta} = x_{f^p}(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) - x_f(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) \tag{20}$$

where f^p is the parent flow of f.

Substituting Eq. (19) into (16), (20) into (17), we have

$$\mu_l^\alpha(t+1) = [\mu_l^\alpha(t) + \gamma(\sum_{f \in \mathcal{F}(l)} x_f(\boldsymbol{\mu}^\alpha(t), \boldsymbol{\mu}^\beta(t)) - c_l)]^+ \tag{21}$$

$$\mu_f^\beta(t+1) = [\mu_f^\beta(t) + \gamma(x_f(\boldsymbol{\mu}^\alpha(t), \boldsymbol{\mu}^\beta(t)) - x_{f^p}(\boldsymbol{\mu}^\alpha(t), \boldsymbol{\mu}^\beta(t)))]^+ \tag{22}$$

Eq. (21) 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. (22), 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[2].

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. (10) and (11). Then f can adjust the flow rate x_f by solving Eq. (15).

[2]Eq. (20) and (22) 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. III is only for those flows which have a parent flow

We present our algorithm in Tab. III. Link l and flow f are deemed as entities capable of computing and communicating[3].

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 Send $x_f(t+1)$ to all links $l \in \mathcal{L}(f)$ and all children flows $\{f' \mid f \to f'\}$

TABLE III
ALGORITHM

Let us 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 $\bar{V} = \max_{f'\in\mathcal{F}} V(f')$; $\bar{Z} = \max\{\bar{U}, \bar{V}\}$; $\bar{\kappa} = \max_{f\in\mathcal{F}} \kappa_f$.

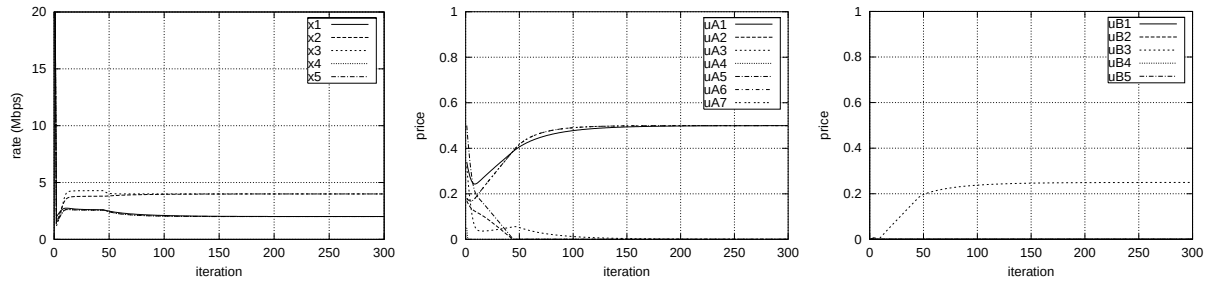
Theorem 1: Assume that $0 < \gamma < 2/\bar{\kappa}\bar{Y}\bar{Z}$, starting from any initial rates $m_f \le x_f(0) \le M_f$, and prices $\boldsymbol{\mu}^\alpha(0) \ge 0$ and $\boldsymbol{\mu}^\beta(0) \ge 0$, every limit point $(\boldsymbol{x}^*, \boldsymbol{\mu}^{\alpha*}, \boldsymbol{\mu}^{\beta*})$ of the sequence $(\boldsymbol{x}(t), \boldsymbol{\mu}^\alpha(t), \boldsymbol{\mu}^\beta(t))$ generated by the algorithm in Tab. III is primal-dual optimal.

The proof is given in the Appendix.

D. 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 time-varying values of flow rates, link prices and relay prices are plotted in Fig. 2. The resulting optimal rates are $x_1^* = 2$, $x_2^* = 4$, $x_3^* = 4$, $x_4^* = 2$, $x_5^* = 2$. The aggregate utility is $\sum_{f\in\mathcal{F}} U_f(x_f^*) = 4.852$.

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



(a) Flow Rates x_f ($f \in \mathcal{F}$) (b) Link Price μ_l^α ($l \in \mathcal{L}$) (c) Relay Price μ_f^β ($f \in \mathcal{F}$)

Fig. 2. Example illustrating the Algorithm

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 temporarally remove constraint (6) in problem $\mathbf{P}$. Consequently, the relay price vector $\boldsymbol{\mu}^\beta$ is removed from the Lagrangian form (8). $\boldsymbol{\lambda}^\beta$ is also removed. In fact, the problem falls back to the unicast flow rate allocation, whose details can be found at [8]. Reflected in the algorithm, the rate adaptation function is modifed 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.

IV. Distributed Asynchronous Algorithm

So far, our algorithm has assumed that the flow rate updates and link/relay price updates are synchronized within the entire overlay session at times $t = 1, 2, \ldots$. In realistic network environment, however, such synchronization is extremely expensive, if not at all impossible, to maintain. In this section, we improve the algorithm to an asynchronous setting, where different links or flows update their rates or prices at different times.

A. Asynchronous Model

We first introduce the asynchronous model used by our algorithm. Let $\mathcal{T} = \{0, 1, 2, \ldots\}$ be the set of time instances at which either a flow rate or a link/relay price is updated. We define

1) $\mathcal{T}_f \subseteq \mathcal{T}$ – the set of time instances at which a flow f updates its rate x_f
2) $\mathcal{T}_l^\alpha \subseteq \mathcal{T}$ – the set of time instances at which a link l updates its link price μ_l^α
3) $\mathcal{T}_f^\beta \subseteq \mathcal{T}$ – the set of time instances at which a flow f updates its relay price μ_f^β.

We further make the following assumption.

– **A4**. There exists a positive integer T such that (1) for every flow f and link l, the time between consecutive updates is bounded by T for both price and rate updates; (2) one-way communication delay between any two entities (links or flows) is at most T time units.

B. Link Price Update

In the asynchronous model, a link l, which updates its price μ_l at time $t \in \mathcal{T}_l^\alpha$, may not have the knowledge of rate information $x_f(t)$ of all flows going through it, i.e., $f \in \mathcal{F}(l)$. Instead, it only keeps track of all recent rate updates $x_f(t')$, which satisfy $(t - T) \leq t' \leq t$, and calculates f's estimated rate $\hat{x}_f^l(t)$ by using a weighted average of these values:

$$\hat{x}_{fl}(t) = \sum_{t'=t-T}^{t} \rho_{fl}(t', t) x_f(t'), \sum_{t'=t-T}^{t} \rho_{fl}(t', t) = 1 \tag{23}$$

Then, link l computes its price according to

$$\mu_l^\alpha(t+1) = [\mu_l^\alpha(t) + \gamma(\sum_{f \in \mathcal{F}(l)} \hat{x}_{fl}(t) - c_l)]^+, \forall t \in \mathcal{T}_l^\alpha \tag{24}$$

Note that at all times $t \notin \mathcal{T}_l^\alpha$, μ_l stays unchanged, i.e., $\mu_l^\alpha(t+1) = \mu_l^\alpha(t)$.

C. Relay Price Update

The relay price is updated in the similar way. A flow f collects all recent rate updates of its parent flow $x_{f^p}(t')$, which satisfy $(t - T) \leq t' \leq t$, and calculates its estimated rate $\hat{x}_{f^p f}(t)$ by using a weighted average of these values:

$$\hat{x}_{f^p f}(t) = \sum_{t'=t-T}^{t} \rho_{f^p f}(t', t) x_f(t'), \sum_{t'=t-T}^{t} \rho_{f^p f}(t', t) = 1 \tag{25}$$

Then, f computes its price according to

$$\mu_f^\beta(t+1) = [\mu_f^\beta(t) + \gamma(x_f(t) - \hat{x}_{f^p f}(t))]^+, \forall t \in \mathcal{T}_f^\beta \tag{26}$$

Note that at all times $t \notin \mathcal{T}_f^\beta$, μ_f^β stays unchanged, i.e., $\mu_f^\beta(t+1) = \mu_f^\beta(t)$.

D. Flow Rate Update

To update the flow rate, a flow f needs to first acquire the estimated prices $\hat{\mu}_{lf}^\alpha(t)$ of all links it goes through, i.e., $l \in \mathcal{L}(f)$, and the estimated prices $\hat{\mu}_{f'f}^\beta(t)$ of all its children flows, i.e., $f \to f'$.

$\hat{\mu}_{lf}^\alpha(t)$ is calculated as the weighted average of recent updates of $\mu_l^\alpha(t')(t - T \le t' \le t)$:

$$\hat{\mu}_{lf}^\alpha(t) = \sum_{t'=t-T}^{t} \rho_{lf}^\alpha(t', t)\mu_l^\alpha(t'), \sum_{t'=t-T}^{t} \rho_{lf}^\alpha(t', t) = 1 \tag{27}$$

$\hat{\mu}_{f'f}^\beta(t)$ is calculated as the weighted average of recent updates of $\mu_{f'}^\beta(t')(t - T \le t' \le t)$:

$$\hat{\mu}_{f'f}^\beta(t) = \sum_{t'=t-T}^{t} \rho_{f'f}^\beta(t', t)\mu_{f'}^\beta(t'), \sum_{t'=t-T}^{t} \rho_{f'f}^\beta(t', t) = 1 \tag{28}$$

Then, flow f computes its rate according to:

$$x_f(t+1) = [U_f'^{-1}(\hat{\lambda}_f^\alpha(t) + \hat{\lambda}_f^\beta(t))]_{m_f}^{M_f}, \forall t \in \mathcal{T}_f \tag{29}$$

where $\hat{\lambda}_f^\alpha(t) = \sum_{l \in \mathcal{L}(f)} \hat{\mu}_{lf}^\alpha(t)$, and $\hat{\lambda}_f^\beta(t) = \mu_f^\beta(t) - \sum_{f \to f'} \hat{\mu}_{f'f}^\beta(t)$. Note that at all times $t \notin \mathcal{T}_f$, x_f stays unchanged, i.e., $x_f(t+1) = x_f(t)$.

E. Algorithm

We present the asynchronous algorithm in Tab. III. Link l and flow f are deemed as entities capable of computing and communicating[4]. In this algorithm, the elements of $\mathcal{T}$ can be viewed as the indices of the sequence of physical times at which updates to either prices or rates occur. The sets $\mathcal{T}_f$, $\mathcal{T}_l^\alpha$, $\mathcal{T}_f^\beta$ as well as the physical times they represent need not be known to any other nodes, since their knowledge is not required in the price and rate computation. Thus, there is no requirement for synchronizing the local clocks at different nodes.

Theorem 2: Assume that the stepsize γ is sufficiently small, 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 limit 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. V is primal-dual optimal.

One issue remained is how to determine the weighted values $\rho_{fl}(t', t)$, $\rho_{fpf}(t', t)$, $\rho_{lf}^\alpha(t', t)$ and $\rho_{f'f}^\beta(t', t)$. Note that the proof of **Theorem 2** does not rely any presumption of what values they should take. In our experiment, we try the following update policies.

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

Notation	Definition
$\mathcal{T} = \{0, 1, 2, \ldots\}$	Set of All Time Instances
$\mathcal{T}_f \in \mathcal{T}$ $(f \in \mathcal{F})$	Set of Time Instances when f Updates its rate x_f
$\mathcal{T}_l^\alpha \in \mathcal{T}$ $(l \in \mathcal{L})$	Set of Time Instances when l Updates its link price μ_l^α
$\mathcal{T}_f^\beta \in \mathcal{T}$ $(f \in \mathcal{F})$	Set of Time Instances when f Updates its relay rate μ_f^β
T	Update Delay Bound
$\hat{x}_{fl}(t)$	Estimation of x_f by link l at time t
$\rho_{fl}(t', t)$	Weighted Value to Assist the Computation of $\hat{x}_{fl}(t)$
$\hat{x}_{f^p f}(t)$	Estimation of x_{f^p} by its child flow f at time t
$\rho_{f^p f}(t', t)$	Weighted Value to Assist the Computation of $\hat{x}_{f^p f}(t)$
$\hat{\mu}_{lf}^\alpha(t)$	Estimation of μ_l^α by flow f at time t
$\rho_{lf}^\alpha(t', t)$	Weighted Value to Assist the Computation of $\hat{\mu}_{lf}^\alpha(t)$
$\hat{\mu}_{f'f}^\beta(t)$	Estimation of $\mu_{f'}^\beta$ by its parent flow f at time t
$\rho_{f'f}^\beta(t', t)$	Weighted Value to Assist the Computation of $\hat{\mu}_{f'f}^\beta(t)$

TABLE IV

NOTATIONS IN SEC. IV

Link Price Update (by link l): At times $t \in \mathcal{T}_l^\alpha$

1 Receive rates $x_f(t')$ from all flows $f \in \mathcal{F}(l)$ from time to time, and keep $x_f(t')$ for $(t - T) \leq t' \leq t$
2 Estimate rate $\hat{x}_{fl}(t)$ according to Eq. (23)
3 Compute price $\mu_l^\alpha(t + 1)$ according to Eq. (24)
4 Send $\mu_l^\alpha(t + 1)$ to all flows $f \in \mathcal{F}(l)$

Relay Price Update (by flow f): At times $t \in \mathcal{T}_f^\beta$

1 Receive rate $x_{f^p}(t')$ from its parent flow f^p from time to time, and keep $x_{f^p}(t')$ for $(t - T) \leq t' \leq t$
2 Estimate rate $\hat{x}_{f^p f}(t)$ according to Eq. (25)
3 Compute price $\mu_f^\beta(t + 1)$ according to Eq. (26)
4 Send $\mu_f^\beta(t + 1)$ to f^p

Stream Rate Adaptation (by flow f): At times $t \in \mathcal{T}_f$

1 Receive link prices $\mu_l^\alpha(t')$ from all links $l \in \mathcal{L}(f)$ from time to time, and keep $\mu_l^\alpha(t')$ for $(t - T) \leq t' \leq t$
2 Estimate link price $\hat{\mu}_{lf}^\alpha(t)$ according to Eq. (27)
3 Receive relay prices $\mu_{f'}^\beta(t')$ from all flows $f \to f'$ from time to time, and keep $\mu_{f'}^\beta(t')$ for $(t - T) \leq t' \leq t$
4 Estimate link price $\hat{\mu}_{f'f}^\beta(t)$ according to Eq. (28)
5 Calculate $\hat{\lambda}_f^\alpha(t) = \sum_{l \in \mathcal{L}(f)} \hat{\mu}_{lf}^\alpha(t)$ and $\hat{\lambda}_f^\beta(t) = \mu_f^\beta(t) - \sum_{f \to f'} \hat{\mu}_{f'f}^\beta(t)$
6 Compute rate $x_f(t + 1)$ according to Eq. (29)
7 Send $x_f(t + 1)$ to all links $l \in \mathcal{L}(f)$ and all children flows $\{f' \mid f \to f'\}$

TABLE V

ASYNCHRONOUS ALGORITHM

1) Latest Update Only – only the last received flow rate update $x_f(t)$ is used to estimate $\hat{x}_{fl}(t)$, i.e., $\rho_{fl}(t,t) = 1$ and other weighted values are set to be 0. The same policy applies for $\rho_{fpf}(t',t)$, $\rho^{\alpha}_{lf}(t',t)$ and $\rho^{\beta}_{f'f}(t',t)$.
2) Latest Average – If there are k updates within the last T time units, then $\rho_{fl}(t',t) = 1/k$ for $t - T \leq t' \leq t$. The same policy applies for $\rho_{fpf}(t',t)$, $\rho^{\alpha}_{lf}(t',t)$ and $\rho^{\beta}_{f'f}(t',t)$.

V. Protocol Design and Implementation

The algorithm presented in Sec. III-C 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[8][9], 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. However, this assumption is not valid in the context of overlay network, whose fundamental design objective is to leave the existing infrastructure unchanged. Therefore, our protocol design and implementation should purely depend on the coordination of end hosts.

A. 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 each end host h is connected to only one router, i.e., it has only one access link. Thus, this link is shared by all flows originated and terminated at h. Our observation is that most end hosts today have only one activated network interface to the Internet.

Third, we assume that the underlying route of a flow path can be found by network path finding tools such as *traceroute*. This enables a receiver to have explicit knowledge of what physical links are passed from the sender to itself. In our implementation, each receiver updates its flow route information to the server upon joining the overlay multicast, or when the route is changed. Considering the fact that most routes in Internet today is relatively stable[13], the update overhead is small.

Our final assumption is that the available bandwidth of each physical link can be measured by tools such as *pathchar*[14] and *pathrate*[15], in an end-to-end manner. Note that we do not need to measure each individual physical link separately. For two adjacent links l and l', if they are shared by the same set of flows ($\mathcal{F}(l) = \mathcal{F}(l')$), they can be seen as one link from end-to-end perspective. The available bandwidth of this link is the smallest available bandwidth of l and l'. This "merging" process generally applies for the case of a chain of links.

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
$\mathbb{P}_h$ ($h \in \mathcal{H}$)	The Set Storing $\mathbf{LPU}$ messages
$\mathbb{R}_h$ ($h \in \mathcal{H}$)	The Set Storing $\mathbf{FRR}$ messages
$t_h^{\mathbb{P}}$ ($h \in \mathcal{H}$)	The time for next $\mathbf{LPU}$ update
$W_h^{\mathbb{P}}$ ($h \in \mathcal{H}$)	The interval between consecutive $\mathbf{LPU}$ updates
$t_h^{\mathbb{R}}$ ($h \in \mathcal{H}$)	The time for next $\mathbf{FRR}$ update
$W_h^{\mathbb{R}}$ ($h \in \mathcal{H}$)	The interval between consecutive $\mathbf{FRR}$ updates

TABLE VI
NOTATIONS IN SEC. V

B. Protocol

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. Now, since each link l or flow f is assigned to an end host, we see that assumption **A4** in Sec. IV can be satisfied if the one-way communication delay between any two end hosts within the overlay multicast session is bounded by T time units. Therefore, the asynchronous algorithm in Tab. V can be directly applied.

Note that to calculate the data price $\lambda_f^\beta(t+1)$, a flow f must know its own relay price $\mu_f^\beta(t)$ and estimated relay prices of its children flows $\hat{\mu}_{f'f}^\beta(t)$ ($f \rightarrow f'$). By Eq. (26), the relay price of f' is calculated based on its own rate $x_{f'}(t)$ and estimated rate of its parent flow $\hat{x}_{ff'}(t)$. This means that each time when f' updates its relay price, it can simply send its rate $x_{f'}(t)$ to its parent flow f, and f is still able to derive its relay price $\mu_{f'}^\beta(t)$ by Eq. (25), if (1) f remembers its own rates in the previous T time units: $x_f(t')$ ($t - T \leq t' \leq t$), and (2) f knows the weights f' assigns to each of them: $\rho_{f'f}^\beta(t', t)$. While the first condition is easily achievable, the second one can also be satisfied if all hosts agree upon a certain update policy as described at the end of Sec. IV. From the flow owner's point of view, O_f can independently calculate $\lambda_f^\beta(t+1)$ if it receives the stream rate reports from all hosts in the set $\mathcal{N}(f)$.

$$\mathcal{N}(f) = O_{fp} \cup \{O_{f'} \mid f \rightarrow f'\} \tag{30}$$

In this way, we remove the need of relay price update. We do so mainly to save messaging overhead. Now we show how it is done. For all flows sharing l, we collect their owners into a set $\mathcal{C}(l)$ as below.

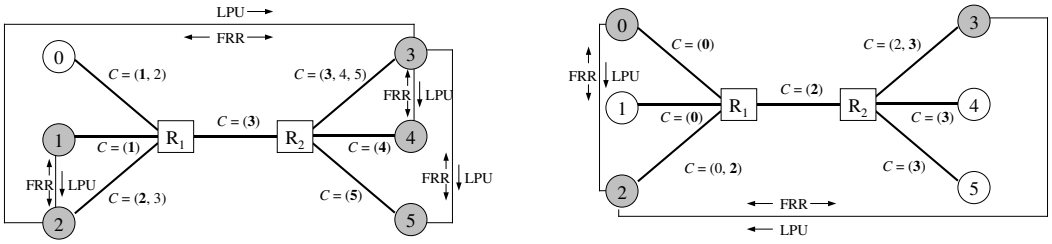
$$\mathcal{C}(l) = \{O_f \mid f \in \mathcal{F}(l)\} \tag{31}$$

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 O_f or D_l in some appropriate way. While O_f is statically assigned to either receiver or sender of f, we make the following rules on choosing D_l.

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. 3, 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. 3 (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. V-A). 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

Fig. 3. Protocol A: Link Delegation

We present the protocol in Tab. VIII. The message formats are listed in Tab. VII. 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.

By the asynchronous algorithm in Tab. V, h also needs to maintain a set containing all time instances at which it sends out **FRR** messages on behalf of all flows it owns, and all time instances at which it sends out **LPU** messages on behalf of all links it delegates. In our protocol, we fix the time interval between consecutive updates of **FRR** and **LPU** messages. Thus, for **LPU** messages, h only needs to maintain $t_h^{\mathbb{P}}$, the time instance of next update, and $W_h^{\mathbb{P}}$, the update interval. Likewise, for **FRR** messages, h only maintains $t_h^{\mathbb{R}}$ and $W_h^{\mathbb{R}}$.

Link Price Update ($\mathbf{LPU}$)		**Flow Rate Report** ($\mathbf{FRR}$)	
$\langle l \rangle$	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
$\langle t \rangle$	Update time	$\langle t \rangle$	Update time

TABLE VII
MESSAGE FORMATS

End Host h

On Receiving $\mathbf{FRR}$ Message

Read $\langle f \rangle$, $\langle x_f \rangle$, $\langle O_f \rangle$ and $\langle t \rangle$ fields of the message
Insert the message into $\mathbb{R}_h$

On Receiving $\mathbf{LPU}$ Message

Read $\langle l \rangle$, $\langle \mu_l^\alpha \rangle$, $\langle D_l \rangle$ and $\langle t \rangle$ fields of the message
Insert the message into $\mathbb{P}_h$

Stream Rate Update: At time $t_h^{\mathbb{R}}$

Remove messages from $\mathbb{R}_h$ and $\mathbb{P}_h$, whose update time is older than $t_h^{\mathbb{R}} - T$
for each f such that $O_f = h$
  Estimate link price $\hat{\mu}_{lf}^\alpha(t_h^{\mathbb{R}})$ for each link $l \in \mathcal{L}(f)$ according to Eq. (27)
  Calculate relay price $\mu_{f'}^\beta(t')$ ($t_h^{\mathbb{R}} - T \le t' \le t_h^{\mathbb{R}}$) for each flow f' ($f \to f'$) according to Eq. (26)
  Derive relay price estimation $\hat{\mu}_{f'f}^\beta(t_h^{\mathbb{R}})$ for each flow f' ($f \to f'$) according to Eq. (27)
  Calculate $\hat{\lambda}_f^\alpha(t_h^{\mathbb{R}}) = \sum_{l \in \mathcal{L}(f)} \hat{\mu}_{lf}^\alpha(t_h^{\mathbb{R}})$ and $\hat{\lambda}_f^\beta(t_h^{\mathbb{R}}) = \mu_f^\beta(t_h^{\mathbb{R}}) - \sum_{f \to f'} \hat{\mu}_{f'f}^\beta(t_h^{\mathbb{R}})$
  Compute rate $x_f(t_h^{\mathbb{R}} + 1)$ according to Eq. (29)
  Send $\mathbf{FRR}$ message to all hosts in $\{D_l \mid l \in \mathcal{L}(f)\} \cup \mathcal{N}_f$,
   setting $\langle f \rangle \leftarrow f$, $\langle x_f \rangle \leftarrow x_f(t_h^{\mathbb{R}} + 1)$, $\langle O_f \rangle \leftarrow h$, $\langle t \rangle \leftarrow t_h^{\mathbb{R}} + 1$
$t_h^{\mathbb{R}} \leftarrow t_h^{\mathbb{R}} + W_h^{\mathbb{R}}$

Link Price Update: At time $t_h^{\mathbb{P}}$

Remove messages from $\mathbb{R}_h$ and $\mathbb{P}_h$, whose update time is older than $t_h^{\mathbb{R}} - T$
for each l such that $D_l = h$
  Estimate rate $\hat{\mu}_{fl}(t_h^{\mathbb{P}})$ for each flow $f \in \mathcal{F}(l)$ according to Eq. (23)
  Compute price $\mu_l^\alpha(t_h^{\mathbb{P}} + 1)$ according to Eq. (24)
  Send $\mathbf{LPU}$ message to all hosts in $\mathcal{C}(l)$,
   setting $\langle l \rangle \leftarrow l$, $\langle \mu_l^\alpha \rangle \leftarrow \mu_l^\alpha(t_h^{\mathbb{P}} + 1)$, $\langle D_l \rangle \leftarrow h$, $\langle t \rangle \leftarrow t_h^{\mathbb{P}} + 1$
$t_h^{\mathbb{P}} \leftarrow t_h^{\mathbb{P}} + W_h^{\mathbb{P}}$

TABLE VIII
PROTOCOL

C. Discussions

In our protocol, the sender-based version is more efficient than the receiver-based version, in terms of messaging overhead. This is because in overlay multicast, there are fewer senders than receivers. While every flow has a unique receiver, some of them share a common sender. In Fig. 3, the link from host 1 to R_1 is shared by two flows with different receivers (hosts 1 and 2) but the same sender (host 0). Therefore, in sender-based protocol, host 0 has control on both flows, which avoids the message exchange on this link. Comparing Fig. 3 (a) and (b), the number of messages drops from 12 to 6. To further save messaging overhead, we can choose to piggyback the **FRR** or **LPU** messages into data packets or multicast session maintenance messages such as heartbeats. We will show the impact of this implementation choice in Sec. VI.

We also note that our protocol require the receiver/sender of each flow f to be aware of the physical routes of a subset of other flows, which share certain links with f. Our implementation chooses a centralized approach, where the server (host 0 in Fig. 3) collects the physical route information of all flows, then constructs the global topology accordingly. In this way, the server is able to arbitrate the selection of D_l for each link l in *Link Delegation*. However, other distributed publish/subscribe mechanism will also work here, which we consider complementary to this paper.

VI. Simulation Results

A. Experimental Setup

We use the Boston BRITE[16] topology generator to setup our experimental network. We choose the hierarchical topology model, as shown in Fig. 4. 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 overlay node is an end host attached to a single router. The bandwidths of all links in Fig. 4 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, $k = 4$.

B. Flow Rate Convergence

We first test the performance of our solution at converging to the optimal flow rate. We setup an overlay multicast session of 10 members. The multicast tree is shown in Fig. 5. Host 0 is the server. Initially,

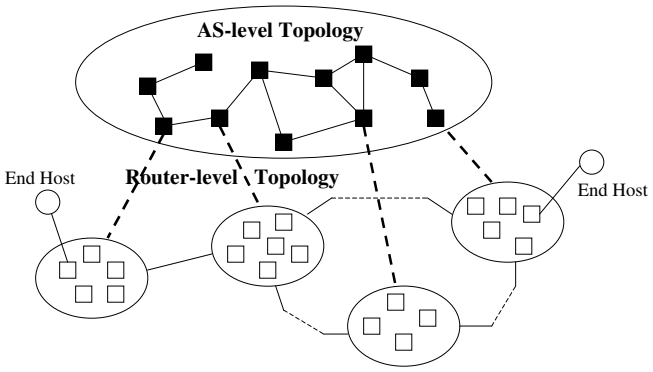


Fig. 4. Experimental Topology

host 1 joins the session. In every minute thereafter, a new member joins. Each member updates is flow rate every 0.1 second. The utility function of every flow f is $U_f(x_f) = \ln(x_f)$. The minimal rate (m_f in Eq. (15)) is 1 Mbps. The maximal rate (M_f in Eq. (15)) is 35 Mbps.

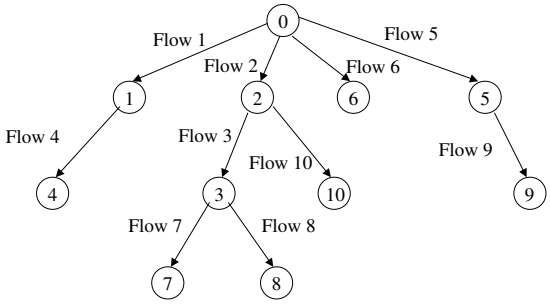
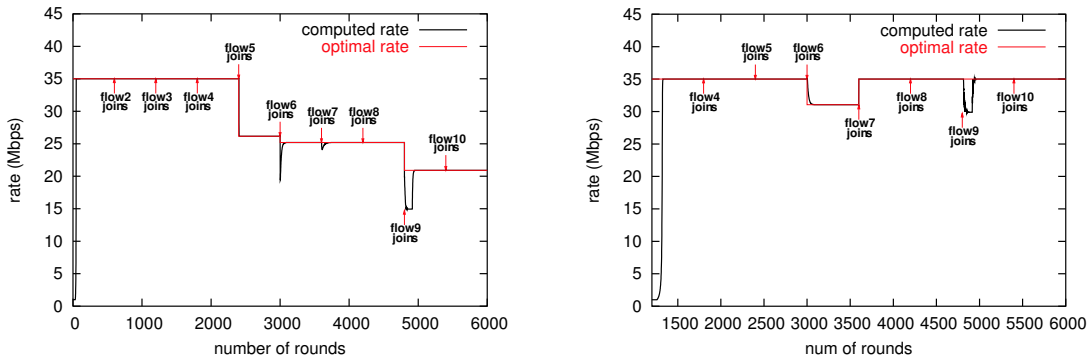


Fig. 5. Experimental Overlay Multicast Tree



(a) Flow 1 (b) Flow 3

Fig. 6. Convergence of Flow Rates (Synchronous Update)

We first show the result of synchronous algorithm (Tab. III), where the step size (γ in Eq. (16) and (17)) is 0.0005. Fig. 6 shows the rate adaptation procedure of Flow 1 and 3. We can see 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 no more than 200 iterations.

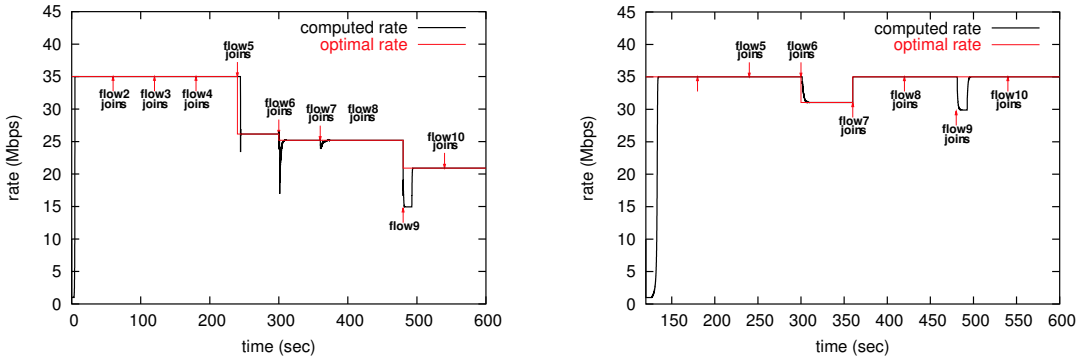
The final optimal rates of all flows are shown in Tab. IX. The aggregate utility is $\sum_{f=1}^{10} U_f(x_f^*) = 32.29$.

Rate($Mbps$)	x_1^*	x_2^*	x_3^*	x_4^*	x_5^*	x_6^*	x_7^*	x_8^*	x_9^*	x_{10}^*
Overlay	20.92	35.00	35.00	20.92	20.92	10.46	35.00	35.00	20.92	35.00
Unicast	21.83	21.83	35.00 (21.83)	25.19 (21.83)	21.83	21.83	35.00 (21.83)	35.00 (21.83)	35.00 (21.83)	35.00 (21.83)

TABLE IX

OPTIMAL RATE COMPARISON OF OVERLAY-BASED AND UNICAST-BASED RESOURCE ALLOCATION SCHEMES

We also compute the optimal rates using the unicast-based resource allocation mechanism reported in [8], 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 30.83, which is suboptimal to the result of overlay-based mechanism.



(a) Flow 1 (Latest Update Only) (b) Flow 3 (Latest Average)

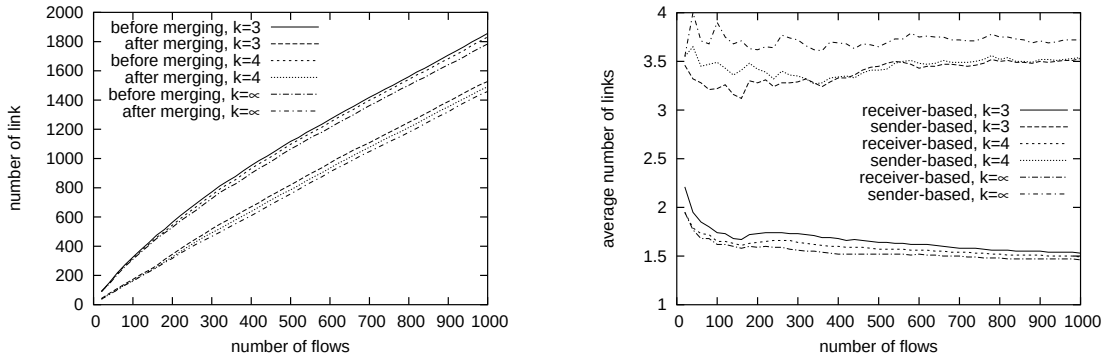
Fig. 7. Convergence of Flow Rates (Asynchronous Update with Average Update Interval of 10 ms, $T = 100$ ms)

We also measure the performance of asynchronous algorithm (Tab. V). Here the stepsize $\gamma = 0.00005$, the average link price and flow rate updates on each node is 10 ms, and the estimation window $T = 100$ ms. First, the algorithm converges to the same optimal rates shown in Tab. IX. Second, as shown in Fig. 7, the rate adaptation procedure of flow 1 (using Latest Update Only policy) and flow 3 (using Latest Average policy) follow the same pattern as in Fig. 6: the curves stay close to the track of the optimal ones, and are disturbed when new members join the multicast session, but quickly converge back to the optimal rates within no more than 20 seconds.

C. Link Measurement Overhead

We now proceed to evaluate the performance of our protocol. One of its task 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.

Fig. 8 (a) shows the number of links the overlay multicast tree contains. This number grows sublinearly when we expand the tree size, because as the number of flows increases, many of them begin to share



(a) Total Number of Links (b) Average Number of Links per Host (After Merging)

Fig. 8. Link Measurement Overhead

some common links. Relaxing the degree constraint (k) also helps to reduce the link number, since more receivers can choose to stream from its closest neighbor, which shortens the flow's physical route. The figure also shows that, when $k = 4$, the link number is already very close to the unconstrained case ($k = \infty$). Finally, when we adopt the "link merging" approach (introduced in Sec. V-A) by treating adjacent links sharing the same set of flows as one link, the number of links can be further reduced. The reduction factor gradually dwindles from 50% (100 flows) to 23% (1000 flows) as the multicast session expands. The reason for this diminishing return is that, when the network is saturated by more flows, the flow set of each link becomes more diversified, which makes it less likely for two adjacent links to happen to share exactly the same set of flows.

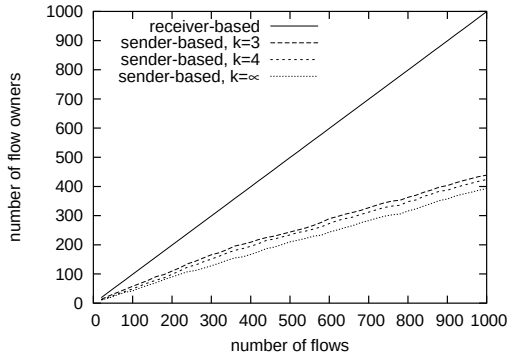


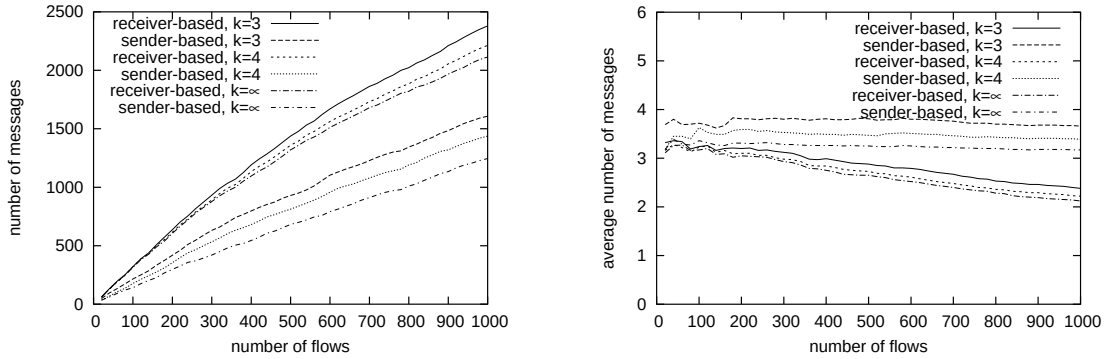
Fig. 9. Number of Flow Owners

Fig. 8 (b) shows the average link measurement overhead per host, where the receiver-based protocol exhibits great scalability. The average number of link measurement operations per receiver slightly decreases as the multicast session expands. This phenomenon corresponds to the sublinear growth of link number in Fig. 8 (a). However, for the sender-based protocol, 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 lies in Fig. 9. Since there 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.

D. Messaging Overhead

Another task of our protocol 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.



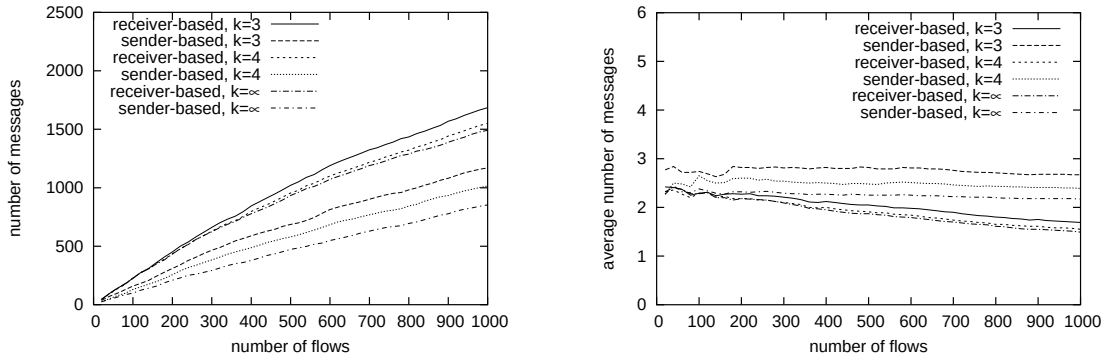
(a) Total Overhead (b) Average Overhead per Host

Fig. 10. Messaging Overhead

Fig. 10 (a) shows the overall messaging overhead, the total number of messages sent out in one round of flow rate adaptation. Note that since our protocol is asynchronous, we cannot strictly indicate as which message belongs to which round. Instead, we measure the messaging overhead in a larger time scale (1000 times the average update interval), then take the average overhead per interval as the result. We observe that the sender-based approach is more efficient than the receiver-based approach. This observation can be illustrated by the same example in Sec. VI-C. 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. Fig. 10 (b) shows that the average messaging overhead per host remains stable as the size of the multicast session grows.

Fig. 11 shows the messaging overhead when we piggyback the message packet into the data packet if two of them share the same source and destination. Compared to the results in Fig. 10, the average message saving is 30%. The lowest message saving is 27% for receiver-based protocol when $k = 3$. The highest message saving is 31% for sender-based protocol when $k = \infty$.

In both Fig. 10 and Fig. 11, we find out that increasing k results in less messaging overhead, since it helps reduce the number of links in a multicast session (Fig. 8 (a)), which in turn helps reduce the total



(a) Total Overhead (b) Average Overhead per Host

Fig. 11. Messaging Overhead (With Data Packet Piggybacking)

number of messages for link price calculation.

We conclude the experimental results of Sec. VI-C and VI-D as follows. First, 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, regarding the link measurement overhead. Second, the protocol overhead is affected by the way the multicast tree is built: increasing k results in fewer number of network links in a multicast session, thus fewer messages are required for link price calculation. However, doing so might make a host to have too many children, which becomes the bottleneck to further increase the streaming rate. Finally, piggybacking messages to data packets also help to reduce significant messaging overhead for both sender-based and receiver-based protocols.

VII. Related Work

Due to the difficulty of deployment of IP multicast, algorithms promoting application-layer overlay multicast have recently been proposed as remedial solutions, focusing on the issue of constructing and maintaining a multicast tree. The common objective is to perform multicast with only unicasts between end hosts, and to minimize the inefficiency brought forth by link stress and stretch. Narada [1], for example, constructs trees in a two-step process: it first constructs an efficient mesh among members, and in the second step construct a spanning tree of the mesh. More recently, researchers have focused on designing scalable overlay tree construction algorithms, using tools including Delaunay Triangulations [17] and organizing members into hierarchies of clusters [18]. Soon realizing the tremendous potentials of

such a new communication paradigm, many studies start to extend its usage into a variety of network and application designs, such as data/service indirection [19], resilient routing [20], and peer-to-peer streaming [21], etc. Our study further expands the boundary by proposing to achieve multi-rate multicast streaming in the setting overlay network.

The price-based resouarce allocation strategies have been extensively explored in the context of IP unicast and multicast. In [6] and [7], 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.[8] then presents a distributed algorithm based on the dual approach of the same problem. In their follow-up work[22], they suggest a randomized marking based implementation of the algorithm in [8], which uses only a bit for the network congestion feedback. 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[11]. Then in [10], they propose a low-overhead implementation of the algorithm, which associates a congestion bit with each link to replace the explicit price signal.

The fundamental difference of our work to the above ones, as we have argued in Sec. I, 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 obviously is an unrealistic assumption in the context of overlay network. For the purpose of practicability, our protocols are designed to purely depend on the coordination of end hosts.

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. [23] presents a global flow control scheme to manage overlay resources, including bandwidth and buffer space of overlay routers. Opus[24] 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[25][4] have explored the optimal utilization of end host buffer spaces to facilitate overlay-based multimedia distribution.

VIII. Concluding Remarks

In this paper, we target 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 due to the dual role of end hosts as both receiver and senders. Practically, our solution has to be purely end-host-based in accordance with the design objective of overlay network. With respect to these

challenges, we propose a distributed algorithm, which maximizes the aggregate utility of all multicast members, subject to both network and data constraints. We then implement our algorithm in a series of protocols purely depending on the coordination of end hosts. Our experiments prove the scalability and efficiency of our solution.

REFERENCES

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

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

[3] D. Rubenstein, J. Kurose and M. Vetterli, "The impact of multicast layering on network fairness," in *ACM SIGCOMM*, 1999.

[4] Y. Cui and K. Nahrstedt, "Layered peer-to-peer streaming," in *NOSSDAV*, 2003.

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

[6] F. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, no. 1, 1997.

[7] F. Kelly, A. Maulloo and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *Journal of Operations Research Society*, vol. 49, no. 3, 1998.

[8] S. Low and D. Lapsley, "Optimization flow control, i: Basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, 1999.

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

[10] 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.

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

[12] D. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1995.

[13] V. Paxson, "End-to-end routing behavior in the internet," in *ACM SIGCOMM*, 1996.

[14] V. Jacobson, *Pathchar*, http://www.caida/org/tools/utilities/others/pathchar.

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

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

[17] J. Liebeherr, M. Nahas and W. Si, "Application-Layer Multicasting With Delaunay Triangulation Overlays," *IEEE Journal on Selected Areas in Communications*, pp. 1472–1488, 2002.

[18] S. Banerjee, B. Bhattacharjee and C. Kommareddy, "Scalable Application Layer Multicast," in *Proc. of ACM SIGCOMM*, August 2002.

[19] I. Stoica, D. Adkins, S. Zhuang, S. Shenker and S. Surana, "Internet Indirection Infrastructure," in *Proc. of ACM SIGCOMM*, August 2002.

[20] D. Anderson, H. Balakrishnan, M. Kaashoek and R. Morris, "Resilient Overlay network," in *Proc. of ACM SOSP*, 2001.

[21] V. Padmanabhan, H. Wang, P. Chou and K. Sripanidkulchai, "Distributing Streaming Media Content using Cooperative Networking," in *Proc. of ACM NOSSDAV*, 2002.

[22] D. Lapsley and S. Low, "Random early marking for internet congestion control," in *IEEE GLOBECOMM*, 1999.

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

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

[25] 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.

IX. Appendix: Proof of Theorem 1

Lemma 1: Let us define a vector $\boldsymbol{\mu} \triangleq (\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta})$, which includes prices of all links $l \in \mathcal{L}$ and flows $f \in \mathcal{F}$. Under assumption A1, the dual objective function $D(\boldsymbol{\mu})$ is convex, lower bounded, and continuously differentiable[5].

Proof: For any price vector $\boldsymbol{\mu}$ define $\psi_f(\boldsymbol{\mu})$ as

$$\psi_f(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta}) = \begin{cases} \frac{1}{-U_f''((x_f(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta})))} & \text{if } U_f'(M_f) \leq \lambda_f^{\alpha} + \lambda_f^{\beta} \leq U_f'(m_f) \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

where λ_f^{α} and λ_f^{β} are defined as in (10), (11) and x_f is defined as in (15).

Now we define $\boldsymbol{H}(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta}) = diag(\psi_f(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta}), f \in \mathcal{F})$ be a $|\mathcal{F}| \times |\mathcal{F}|$ diagonal matrix with diagonal elements $\psi_f(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta})$. By assumption A2, we have $0 \leq \psi_f(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta}) \leq \kappa_f$. Define a $(L+F) \times (L+F)$ matrix as follows. We have $\nabla^2 D(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta}) = \boldsymbol{R}(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta})$.

$$\boldsymbol{R}(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta}) = \begin{pmatrix} \boldsymbol{A}\boldsymbol{H}(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta})\boldsymbol{A}^T & \boldsymbol{A}\boldsymbol{H}(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta})\boldsymbol{B}^T \\ \boldsymbol{B}\boldsymbol{H}(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta})\boldsymbol{A}^T & \boldsymbol{B}\boldsymbol{H}(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta})\boldsymbol{B}^T \end{pmatrix}$$

■

Lemma 2: Under assumption A1, the Hessian of D is given by $\nabla^2 D(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta}) = \boldsymbol{R}(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta})$, where it exists.

Proof: Let $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}^{\alpha}}(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta})$ denote the $|\mathcal{F}| \times |\mathcal{L}|$ Jacobian matrix whose (f, l) element is $(\frac{\partial x_f(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta})}{\partial \mu_l^{\alpha}})$, where

$$\frac{\partial x_f(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta})}{\partial \mu_l^{\alpha}} = \begin{cases} \frac{A_{lf}}{U_f''((x_f(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta})))} & \text{if } U_f'(M_f) \leq \lambda_f^{\alpha} + \lambda_f^{\beta} \leq U_f'(m_f) \\ 0 & \text{otherwise} \end{cases}$$

Let $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}^{\beta}}(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta})$ denote the $|\mathcal{F}| \times |\mathcal{F}|$ Jacobian matrix whose (f, f') element is $(\frac{\partial x_f(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta})}{\partial \mu_{f'}^{\beta}})$, where

$$\frac{\partial x_f(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta})}{\partial \mu_{f'}^{\beta}} = \begin{cases} \frac{B_{f'f}}{U_f''((x_f(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta})))} & \text{if } U_f'(M_f) \leq \lambda_f^{\alpha} + \lambda_f^{\beta} \leq U_f'(m_f) \\ 0 & \text{otherwise} \end{cases}$$

From (19), (20) we have,

$$\begin{aligned} \nabla_{\boldsymbol{\mu}^{\alpha}} D(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta}) &= \boldsymbol{C} - \boldsymbol{A}\boldsymbol{x}(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta}) \\ \nabla_{\boldsymbol{\mu}^{\beta}} D(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta}) &= \boldsymbol{B}\boldsymbol{x}(\boldsymbol{\mu}^{\alpha}, \boldsymbol{\mu}^{\beta}) \end{aligned}$$

[5]This is different from the IP multicast case[9]

Using (32), we have

$$[\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}^{\alpha}}(\boldsymbol{\mu}^{\alpha},\boldsymbol{\mu}^{\beta})] = -\boldsymbol{H}(\boldsymbol{\mu}^{\alpha},\boldsymbol{\mu}^{\beta})\boldsymbol{A}^{T}$$

$$[\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}^{\beta}}(\boldsymbol{\mu}^{\alpha},\boldsymbol{\mu}^{\beta})] = -\boldsymbol{H}(\boldsymbol{\mu}^{\alpha},\boldsymbol{\mu}^{\beta})\boldsymbol{B}^{T}$$

Thus,

$$\nabla^{2}_{\boldsymbol{\mu}^{\alpha}\boldsymbol{\mu}^{\alpha}} D(\boldsymbol{\mu}^{\alpha},\boldsymbol{\mu}^{\beta}) = -\boldsymbol{A}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}^{\alpha}}(\boldsymbol{\mu}^{\alpha},\boldsymbol{\mu}^{\beta}) = \boldsymbol{A}\boldsymbol{H}(\boldsymbol{\mu}^{\alpha},\boldsymbol{\mu}^{\beta})\boldsymbol{A}^{T}$$

$$\nabla^{2}_{\boldsymbol{\mu}^{\beta}\boldsymbol{\mu}^{\beta}} D(\boldsymbol{\mu}^{\alpha},\boldsymbol{\mu}^{\beta}) = -\boldsymbol{B}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}^{\beta}}(\boldsymbol{\mu}^{\alpha},\boldsymbol{\mu}^{\beta}) = \boldsymbol{B}\boldsymbol{H}(\boldsymbol{\mu}^{\alpha},\boldsymbol{\mu}^{\beta})\boldsymbol{B}^{T}$$

$$\nabla^{2}_{\boldsymbol{\mu}^{\alpha}\boldsymbol{\mu}^{\beta}} D(\boldsymbol{\mu}^{\alpha},\boldsymbol{\mu}^{\beta}) = -\boldsymbol{A}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}^{\beta}}(\boldsymbol{\mu}^{\alpha},\boldsymbol{\mu}^{\beta}) = \boldsymbol{A}\boldsymbol{H}(\boldsymbol{\mu}^{\alpha},\boldsymbol{\mu}^{\beta})\boldsymbol{B}^{T}$$

$$\nabla^{2}_{\boldsymbol{\mu}^{\beta}\boldsymbol{\mu}^{\alpha}} D(\boldsymbol{\mu}^{\alpha},\boldsymbol{\mu}^{\beta}) = -\boldsymbol{B}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}^{\alpha}}(\boldsymbol{\mu}^{\alpha},\boldsymbol{\mu}^{\beta}) = \boldsymbol{B}\boldsymbol{H}(\boldsymbol{\mu}^{\alpha},\boldsymbol{\mu}^{\beta})\boldsymbol{A}^{T}$$

■

Lemma 3: Under assumption A1 and A2, we have ∇D is Lipschitz with

$$||\nabla D(q) - \nabla D(p)||_2 \leq \kappa \bar{Y}\bar{Z}||q-p||_2 \tag{33}$$

for all $p, q \geq 0$. *Proof:* Given any $p, q \geq 0$, using Taylor theorem and Lemma 2 we have

$$\nabla D(q) - \nabla D(p) = \nabla^2 D(w)(q-p) = \boldsymbol{R}(w)(q-p)$$

for some $w = tp + (1-t)q \geq 0$, $t \in [0,1]$. Hence,

$$||\nabla D(q) - \nabla D(p)||_2 \leq ||\boldsymbol{R}(w)||_2 \cdot ||q-p||_2$$

Now we show that $||\boldsymbol{R}(w)||_2 \leq \kappa \bar{Y}\bar{Z}$.

First

$$||\boldsymbol{R}(w)||_2^2 \leq ||\boldsymbol{R}(w)||_\infty \cdot ||\boldsymbol{R}(w)||_1$$

Since $\boldsymbol{R}(w)$ is symmetric, we have

$$||\boldsymbol{R}(w)||_\infty = ||\boldsymbol{R}(w)||_1$$

Hence,

$$||\boldsymbol{R}(w)||_2 \leq ||\boldsymbol{R}(w)||_\infty = \max_r \sum_{r'} [\boldsymbol{R}(w)]_{rr'}$$

Actually,

$$[\boldsymbol{R}(w)]_{rr'} = \begin{cases} \sum_f \psi_f(w) A_{rf} A_{r'f} & \text{if} \quad r, r' \in [0, L-1] \\ \sum_f \psi_f(w) A_{rf} B_{(r'-L)f} & \text{if} \quad r \in [0, L-1], r' \in [L, L+F-1] \\ \sum_f \psi_f(w) B_{(r-L)f} A_{r'f} & \text{if} \quad r \in [L, L+F-1], r' \in [0, L-1] \\ \sum_f \psi_f(w) B_{(r-L)f} B_{(r'-L)f} & \text{if} \quad r, r' \in [L, L+F-1] \end{cases}$$

Now we have,

$$\sum_{r'} [\boldsymbol{R}(w)]_{rr'} = \begin{cases} \sum_f [\psi_f(w) A_{rf} (\sum_{r'=0}^{L-1} A_{r'f} + \sum_{r'=0}^{F-1} B_{r'f})] & \text{if } r \in [0, L-1] \\ \sum_f [\psi_f(w) B_{(r-L)f} (\sum_{r'=0}^{L-1} A_{r'f} + \sum_{r'=0}^{F-1} B_{r'f})] & \text{if } r \in [L, L+F-1] \end{cases}$$

As $Y(f) = \sum_l A_{lf} + \sum_{f'} B_{r'f}$, and $\bar{Y} = \max_{f \in \mathcal{F}} Y(f)$, we have

$$\sum_{r'} [\boldsymbol{R}(w)]_{rr'} \leq \begin{cases} \bar{Y} \sum_f \psi_f(w) A_{rf} & \text{if } r \in [0, L-1] \\ \bar{Y} \sum_f \psi_f(w) B_{(r-L)f} & \text{if } r \in [L, L+F-1] \end{cases}$$

Also because $U(l) = \sum_{f \in \mathcal{F}} A_{lf}$ and $\bar{U} = \max_{l \in \mathcal{L}} U(l)$; $V(f') = \sum_{f \in \mathcal{F}} B_{f'f}$ and $\bar{V} = \max_{f' \in \mathcal{F}} V(f')$; $\bar{Z} = \max\{\bar{U}, \bar{V}\}$; $\bar{\kappa} = \max_{f \in \mathcal{F}} \kappa_f$, we have

$$\max_r \sum_{r'} [\boldsymbol{R}(w)]_{rr'} \leq \bar{\kappa} \bar{Y} \bar{Z}.$$

■

Thus, the dual objective function D is lower bounded and ∇D is Lipschitz. Then, any accumulation point $(\mu^{\alpha*}, \mu^{\beta*})$ of the sequence $(\mu^\alpha(t), \mu^\beta(t))$ generated by the gradient projection algorithm for the dual problem is dual optimal. Now we can prove **Theorem 1**.

Proof: Let $(\mu^\alpha(t), \mu^\beta(t))$ be a subsequence converging to $(\mu^{\alpha*}, \mu^{\beta*})$. Note that $U_f'(x_f)$ is defined on a compact set $[m_f, M_f]$ and it is continuous and one-to-one. Thus, its inverse is continuous. Also, $x(t)$ is continuous. Because the objective function Eq. (4) is strictly concave, and hence continuous, and the feasible region of Eq. (5) and (6) is compact, there is a unique maximizer x^*. Thus, the subsequence $\{x(t)\}$ converges to the primal optimal rate x^*. ■

APPENDIX B: PROOF OF THEOREM 2

The proof of of Theorem 2 follows similar approach as in [11] (Section 7.5) and [8]. The key to the proof is to show that the price adjustment remains in the descent direction. The proof of our case is more

complicated, because two different types of prices are involved: link price and data price.

Define vector $\boldsymbol{\pi}(t) \triangleq (\boldsymbol{\pi}^{\alpha}(t), \boldsymbol{\pi}^{\beta}(t))$, where $\boldsymbol{\pi}^{\alpha}(t) \triangleq \boldsymbol{\mu}^{\alpha}(t+1) - \boldsymbol{\mu}^{\alpha}(t)$, and $\boldsymbol{\pi}^{\beta}(t) \triangleq \boldsymbol{\mu}^{\beta}(t+1) - \boldsymbol{\mu}^{\beta}(t)$. We first show that the error in rate calculation of flow f is bounded by the successive price change $\boldsymbol{\pi}^{\alpha}$ and $\boldsymbol{\pi}^{\beta}$. Let $\lambda_f(t) = \lambda_f^{\alpha}(t) + \lambda_f^{\beta}(t)$, we have the following lemma.

Lemma 4:

1) For all t

$$|U_f^{'-1}(\hat{\lambda}_f(t)) - U_f^{'-1}(\lambda_f(t))| \leq \kappa_f \sum_{t'=t-T}^{t-1} \Big(\sum_{l \in \mathcal{L}(f)} |\pi_l^{\alpha}(t')| + \sum_{f \to f'} |\pi_{f'}^{\beta}(t')| \Big) \tag{34}$$

2) For all t

$$|U_f^{'-1}(\lambda_f(t)) - U_f^{'-1}(\lambda_f(\tau))| \leq \kappa_f \sum_{t'=\tau}^{t-1} \Big(\sum_{l \in \mathcal{L}(f)} |\pi_l^{\alpha}(t') + \sum_{f \to f'} |\pi_{f'}^{\beta}(t')| \Big) \tag{35}$$

Proof: First let us define $\overline{\mu}_l^{\alpha}(t) \triangleq (\mu_{lt'}^{\alpha}(t), t' \in [t-T, t])$ as a sequence of link l's price at time instances $t-T, t-T+1, ..., t$, $\overline{\mu}_f^{\beta}(t) \triangleq (\mu_{ft'}^{\beta}(t), t' \in [t-T, t])$ as a sequence of flow f's price at time instances $t-T, t-T+1, ..., t$. We further define $\overline{\boldsymbol{\mu}}^{\alpha}(t) \triangleq (\overline{\mu}_l^{\alpha}(t), l \in \mathcal{L})$, $\overline{\boldsymbol{\mu}}^{\beta}(t) \triangleq (\overline{\mu}_f^{\beta}(t), f \in \mathcal{F})$. We then introduce the sequence $\overline{\boldsymbol{\mu}}(t) \triangleq (\overline{\boldsymbol{\mu}}^{\alpha}(t), \overline{\boldsymbol{\mu}}^{\beta}(t))$. Each point in this sequence records the prices of all links and flows at a given time instance t'.

Following the same way as $\overline{\boldsymbol{\mu}}$, we define $\boldsymbol{\epsilon}(t) \triangleq (\boldsymbol{\epsilon}^{\alpha}(t), \boldsymbol{\epsilon}^{\beta}(t))$, where $\boldsymbol{\epsilon}^{\alpha}(t) \triangleq (\epsilon_{lt'}^{\alpha}(t), l \in \mathcal{L}, t' \in [t-T, t])$ is defined as

$$\epsilon_{lt'}^{\alpha}(t) = \begin{cases} \rho_{lf}^{\alpha}(t', t), & \text{if } l \in \mathcal{L}(f) \\ 0, & \text{otherwise} \end{cases}$$

and $\boldsymbol{\epsilon}^{\beta}(t) \triangleq \epsilon_{f't'}^{\beta}(t), f' \in \mathcal{F}, t' \in [t-T, t])$ is defined as

$$\epsilon_{f't'}^{\beta}(t) = \begin{cases} \rho_{f'f}^{\beta}(t', t), & \text{if } f \to f' \\ 0, & \text{otherwise} \end{cases}$$

Now let us define $x_f(\boldsymbol{\epsilon}(t); \overline{\boldsymbol{\mu}}(t))$ as

$$x_f(\boldsymbol{\epsilon}(t); \overline{\boldsymbol{\mu}}(t)) \triangleq U_f^{'-1}\Big(\sum_{t'=t-T}^{t} \Big(\sum_{l \in \mathcal{L}(f)} \epsilon_{lt'}^{\alpha}(t) \mu_{lt'}^{\alpha}(t) + \mu_f^{\beta}(t) - \sum_{f \to f'} \epsilon_{f't'}^{\beta}(t) \mu_{f't'}^{\beta}(t) \Big) \Big) \tag{36}$$

It is easy to see that $U_f^{'-1}(\hat{\lambda}_f^{\alpha}(t) + \hat{\lambda}_f^{\beta}(t)) = x_f(\boldsymbol{\epsilon}(t); \overline{\boldsymbol{\mu}}(t))$. By assumption **A2**, we have that

$$0 \leq \Big| \frac{\partial x_f(\boldsymbol{\epsilon}(t); \overline{\boldsymbol{\mu}}(t))}{\partial \epsilon_{lt'}^{\alpha}(t)} \Big| \leq \kappa_f \mu_l^{\alpha}(t'), l \in \mathcal{L}(f) \tag{37}$$

$$0 \leq \Big| \frac{\partial x_f(\boldsymbol{\epsilon}(t); \overline{\boldsymbol{\mu}}(t))}{\partial \epsilon_{f't'}^{\beta}(t)} \Big| \leq \kappa_f \mu_{f'}^{\beta}(t'), f \to f' \tag{38}$$

where they exist.

Now following the same way as $\boldsymbol{\epsilon}(t)$, we define $\mathbf{1}(t) \triangleq (\mathbf{1}^{\alpha}(t'), \mathbf{1}^{\beta}(t')$, where $\mathbf{1}^{\alpha}(t) \triangleq (1^{\alpha}_{lt'}(t), l \in \mathcal{L}, t' \in [t-T, t])$ is defined as

$$1^{\alpha}_{lt'}(t) = \begin{cases} 1, & \text{if } l \in \mathcal{L}(f) \text{and } t' = t \\ 0, & \text{otherwise} \end{cases}$$

and $\mathbf{1}^{\beta}(t) \triangleq (1^{\beta}_{f't'}(t), f' \in \mathcal{F}, t' \in [t-T, t])$ is defined as

$$1^{\beta}_{f't'}(t) = \begin{cases} 1, & \text{if } f \to f' \text{ and } t' = t \\ 0, & \text{otherwise} \end{cases}$$

It is easy to see that $U_f^{'-1}(\lambda^{\alpha}_f(t) + \lambda^{\beta}_f(t)) = x_f(\mathbf{1}(t); \overline{\boldsymbol{\mu}}(t))$, if $x_f(\mathbf{1}(t); \overline{\boldsymbol{\mu}}(t))$ is defined in the same as in (36).

To prove (34), by the mean value theorem, we have for some $\tilde{\epsilon}$,

$$\begin{aligned}
& |U_f^{'-1}(\hat{\lambda}^{\alpha}_f(t) + \hat{\lambda}^{\beta}_f(t)) - U_f^{'-1}(\lambda^{\alpha}_f(t) + \lambda^{\beta}_f(t))| \\
= & \left| \sum_{t'=t-T}^{t} \left[\sum_{l \in \mathcal{L}(f)} \frac{\partial x_f(\tilde{\epsilon}; \overline{\boldsymbol{\mu}}(t))}{\partial \epsilon^{\alpha}_{lt'}} (1^{\alpha}_{lt'}(t) - \epsilon^{\alpha}_{lt'}(t)) + \sum_{f \to f'} \frac{\partial x_f(\tilde{\epsilon}; \overline{\boldsymbol{\mu}}(t))}{\partial \epsilon^{\beta}_{f't'}} (1^{\beta}_{f't'}(t) - \epsilon^{\beta}_{f't'}(t)) \right] \right| \\
\leq & \kappa_f \left| \sum_{t'=t-T}^{t} \left[\sum_{l \in \mathcal{L}(f)} \overline{\mu}^{\alpha}_{lt'}(t)(1^{\alpha}_{lt'}(t) - \epsilon^{\alpha}_{lt'}(t)) - \sum_{f \to f'} \overline{\mu}^{\beta}_{f't'}(t)(1^{\beta}_{f't'}(t) - \epsilon^{\beta}_{f't'}(t)) \right] \right| \\
\leq & \kappa_f \sum_{l \in \mathcal{L}(f)} \left| \mu^{\alpha}_l(t) - \sum_{t'=t-T}^{t} \rho^{\alpha}_{lf}(t', t) \mu^{\alpha}_l(t') \right| + \kappa_f \sum_{f \to f'} \left| \mu^{\beta}_{f'}(t) - \sum_{t'=t-T}^{t} \rho^{\beta}_{f'f}(t', t) \mu^{\beta}_{f'}(t') \right| \\
\leq & \kappa_f \sum_{l \in \mathcal{L}(f)} \max_{t-T \leq t' \leq t} |\mu^{\alpha}_l(t) - \mu^{\alpha}_l(t')| + \kappa_f \sum_{f \to f'} \max_{t-T \leq t' \leq t} |\mu^{\beta}_{f'}(t) - \mu^{\beta}_{f'}(t')| \\
\leq & \kappa_f \sum_{l \in \mathcal{L}(f)} \max_{t-T \leq t' \leq t} \sum_{\tau=t'}^{t-1} |\pi^{\alpha}_l(\tau)| + \kappa_f \sum_{f \to f'} \max_{t-T \leq t' \leq t} \sum_{\tau=t'}^{t-1} |\pi^{\beta}_{f'}(\tau)| \\
\leq & \kappa_f \sum_{t'=t-T}^{t-1} \left(\sum_{l \in \mathcal{L}(f)} |\pi^{\alpha}_l(t')| + \sum_{f \to f'} |\pi^{\beta}_{f'}(t')| \right)
\end{aligned}$$

To prove (35), also by the mean value theorem, we have

$$\begin{aligned}
& |U_f^{'-1}(\lambda^{\alpha}_f(t) + \lambda^{\beta}_f(t)) - U_f^{'-1}(\lambda^{\alpha}_f(\tau) + \lambda^{\beta}_f(\tau))| \\
\leq & \kappa_f \sum_{l \in \mathcal{L}(f)} |\mu^{\alpha}_l(t) - \mu_l(\tau)| + \kappa_f \sum_{f \to f'} |\mu^{\beta}_{f'}(t) - \mu^{\beta}_{f'}(\tau)| \\
\leq & \kappa_f \sum_{t'=\tau}^{t-1} (\sum_{l \in \mathcal{L}(f)} |\pi^{\alpha}_l(t')| + \sum_{f \to f'} |\pi^{\beta}_{f'}(t')|)
\end{aligned}$$

which completes the proof of **Lemma 4**. ■

The gradient estimation used in our asynchronous algorithm is calculated as

$$\begin{aligned}\xi_l(t) &= c_l - \sum_{f\in\mathcal{F}(l)} \hat{x}_{fl}(t)\\ \xi_f(t) &= \hat{x}_{f^p f}(t) - x_f(t)\end{aligned}$$

Following the same way as $\boldsymbol{\mu}(t)$, we define vector $\boldsymbol{\xi}(t) \triangleq (\boldsymbol{\xi}^\alpha(t), \boldsymbol{\xi}^\beta(t))$, where $\boldsymbol{\xi}^\alpha(t) \triangleq (\xi_l(t), l \in \mathcal{L})$ and $\boldsymbol{\xi}^\beta(t) \triangleq (\xi_f(t), f \in \mathcal{F})$. Next we give the bound of error in gradient estimation in terms of the successive price change $\boldsymbol{\pi}(t)$.

Lemma 5: There exists a constant $K_1 > 0$ such that

$$\|\nabla D(\boldsymbol{\mu}(t)) - \boldsymbol{\xi}(t)\| \le K_1 \sum_{t'=t-2T}^{t-1} \|\boldsymbol{\pi}(t')\| \tag{39}$$

Proof: First,

$$\begin{aligned}(\nabla D(\boldsymbol{\mu}(t)) - \boldsymbol{\xi}(t))_l &= \sum_{f\in\mathcal{F}(l)} \left(\sum_{t'=t-T}^{t} \rho_{fl}(t',t) x_f(t') - \overline{x}_f(t) \right)\\ (\nabla D(\boldsymbol{\mu}(t)) - \boldsymbol{\xi}(t))_f &= \sum_{t'=t-T}^{t} \rho_{f^p f}(t',t) x_{f^p}(t') - \overline{x}_{f^p}(t)\end{aligned}$$

where $\overline{x}_f(t)$ is the rate of flow f if it knows the exact network price $\lambda_f^\alpha(t)$ and relay price $\lambda_f^\beta(t)$. Hence by [11] (Proposition A.2), for some constant $K_1' > 0$ we have

$$\begin{aligned}&\|\nabla D(\boldsymbol{\mu}(t)) - \boldsymbol{\xi}(t)\|\\ \le\ & K_1' \cdot \max\left[\max_{l\in\mathcal{L}} \sum_{f\in\mathcal{F}(l)} \left| \sum_{t'=t-T}^{t} \rho_{fl}(t',t) x_f(t') - \overline{x}_f(t) \right|, \max_{f\in\mathcal{F}} \left| \sum_{t'=t-T}^{t} \rho_{f^p f}(t',t) x_{f^p}(t') - \overline{x}_{f^p}(t) \right| \right]\\ \le\ & K_1' \cdot \max\left[\max_{l\in\mathcal{L}} \sum_{f\in\mathcal{F}(l)} \max_{t-T\le t'\le t} \left| x_f(t') - \overline{x}_f(t) \right|, \max_{f\in\mathcal{F}} \max_{t-T\le t'\le t} \left| x_{f^p}(t') - \overline{x}_{f^p}(t) \right| \right]\\ \le\ & K_1' \cdot \max\left[\max_{l\in\mathcal{L}} \sum_{f\in\mathcal{F}(l)} \max_{t-T\le t'\le t} \left| U_f'^{-1}(\hat{\lambda}_f(t')) - U_f'^{-1}(\lambda_f(t)) \right|, \max_{f\in\mathcal{F}} \max_{t-T\le t'\le t} \left| U_{f^p}'^{-1}(\hat{\lambda}_{f^p}(t')) - U_{f^p}'^{-1}(\lambda_{f^p}(t)) \right| \right]\end{aligned}$$

Applying **Lemma 4**, we have

$$
\begin{aligned}
& \|\nabla D(\boldsymbol{\mu}(t)) - \boldsymbol{\xi}(t)\| \\
\leq\ & K_1' \cdot \max \Bigg[\max_{l\in\mathcal{L}} \sum_{f\in\mathcal{F}(l)} \max_{t-T\leq t'\leq t} \left| U_f'^{-1}(\lambda_f(t)) - U_f'^{-1}(\lambda_f(t')) \right| + \left| U_f'^{-1}(U_f'^{-1}(\lambda_f(t')) - U_f'^{-1}(\hat{\lambda}_f(t')) \right|, \\
& \qquad \max_{f\in\mathcal{F}} \max_{t-T\leq t'\leq t} \left| U_{f^p}'^{-1}(\lambda_{f^p}(t)) - U_{f^p}'^{-1}(\lambda_{f^p}(t')) \right| + \left| U_{f^p}'^{-1}(U_{f^p}'^{-1}(\lambda_{f^p}(t')) - U_{f^p}'^{-1}(\hat{\lambda}_{f^p}(t')) \right| \Bigg] \\
\leq\ & K_1' \cdot \max \\
& \Bigg[\max_{l\in\mathcal{L}} \sum_{f\in\mathcal{F}(l)} \max_{t-T\leq t'\leq t} \kappa_f \Bigg\{ \sum_{\tau=t'}^{t-1} \Big(\sum_{l\in\mathcal{L}(f)} |\pi_l^{\alpha}(\tau)| + \sum_{f\to f'} |\pi_{f'}^{\beta}(\tau)| \Big) + \sum_{\tau=t'-T}^{t'-1} \Big(\sum_{l\in\mathcal{L}(f)} |\pi_l^{\alpha}(\tau)| + \sum_{f\to f'} |\pi_{f'}^{\beta}(\tau)| \Big) \Bigg\}, \\
& \max_{f\in\mathcal{F}} \max_{t-T\leq t'\leq t} \kappa_{f^p} \Bigg\{ \sum_{\tau=t'}^{t-1} \Big(\sum_{l\in\mathcal{L}(f^p)} |\pi_l^{\alpha}(\tau)| + \sum_{f^p\to f'} |\pi_{f'}^{\beta}(\tau)| \Big) + \sum_{\tau=t'-T}^{t'-1} \Big(\sum_{l\in\mathcal{L}(f^p)} |\pi_l^{\alpha}(\tau)| + \sum_{f^p\to f'} |\pi_{f'}^{\beta}(\tau)| \Big) \Bigg\} \Bigg] \\
=\ & K_1' \cdot \max \Bigg[\max_{l\in\mathcal{L}} \sum_{f\in\mathcal{F}(l)} \max_{t-T\leq t'\leq t-1} \kappa_f \Bigg\{ \sum_{\tau=t'-T}^{t-1} \Big(\sum_{l\in\mathcal{L}(f)} |\pi_l^{\alpha}(\tau)| + \sum_{f\to f'} |\pi_{f'}^{\beta}(\tau)| \Big) \Bigg\}, \\
& \qquad \max_{f\in\mathcal{F}} \max_{t-T\leq t'\leq t} \kappa_{f^p} \Bigg\{ \sum_{\tau=t'-T}^{t-1} \Big(\sum_{l\in\mathcal{L}(f^p)} |\pi_l^{\alpha}(\tau)| + \sum_{f^p\to f'} |\pi_{f'}^{\beta}(\tau)| \Big) \Bigg\} \Bigg] \\
\leq\ & K_1' \cdot \max \Bigg[\max_{l\in\mathcal{L}} \sum_{f\in\mathcal{F}(l)} \kappa_f \sum_{\tau=t-2T}^{t-1} \|\boldsymbol{\pi}(\tau)\|_1, \max_{f\in\mathcal{F}} \kappa_{f^p} \sum_{\tau=t-2T}^{t-1} \|\boldsymbol{\pi}(\tau)\|_1 \Bigg] \\
\leq\ & K_1' \bar{\kappa} \bar{Y} \bar{Z} \sum_{\tau=t-2T}^{t-1} \|\boldsymbol{\pi}(\tau)\|_1
\end{aligned}
$$

where the third inequality follows from $\sum_{l\in\mathcal{L}(f)} |\pi_l^{\alpha}(\tau)| + \sum_{f\to f'} |\pi_{f'}^{\beta}(\tau)| \leq \|\boldsymbol{\pi}(\tau)\|_1$, and the last inequality follows from the proof of **Theorem 1**. Note that all norms are equivalent in finite dimensional vector space [11] (Proposition A.9). Let $K_1 = K_1' \bar{\kappa} \bar{Z} \bar{Y}$, then we complete the proof of **Lemma 5**. ■

Now we show that $\|\boldsymbol{\pi}(t)\|$ converges to zero in the following lemma.

Lemma 6: Provided γ is sufficiently small we have $\|\boldsymbol{\pi}(t)\| \to 0$ as $t \to \infty$.

Proof: First, by Lemma 5.1 in [11] (Section 7.5), we have that for all t, $\boldsymbol{\xi}^T(t)\boldsymbol{\pi}(t) \leq -(1/\gamma)\|\boldsymbol{\pi}(t)\|^2$. By the descent lemma [11] (Proposition A.32) and Eq. (33), we have that there exists K_2 such that

$$
\begin{aligned}
& D(\boldsymbol{\mu}(t+1)) \\
\leq\ & D(\boldsymbol{\mu}(t)) + (\nabla D(\boldsymbol{\mu}(t)) - \boldsymbol{\xi}(t))^T \boldsymbol{\pi}(t) + \boldsymbol{\xi}^T(t)\boldsymbol{\pi}(t) + K_2 \|\boldsymbol{\pi}(t)\|^2 \\
\leq\ & D(\boldsymbol{\mu}(t)) + \|\nabla D(\boldsymbol{\mu}(t)) - \boldsymbol{\xi}(t)\| \cdot \|\boldsymbol{\pi}(t)\| - \left(\frac{1}{\gamma} - K_2 \right) \|\boldsymbol{\pi}(t)\|^2
\end{aligned}
$$

Applying **Lemma 5**, we have

$$\begin{aligned} & D(\boldsymbol{\mu}(t+1)) \\ \leq\ & D(\boldsymbol{\mu}(t)) - \left(\frac{1}{\gamma} - K_2\right)\|\boldsymbol{\pi}(t)\|^2 + K_1 \sum_{t'=t-2T}^{t-1} \|\boldsymbol{\pi}(t')\| \cdot \|\boldsymbol{\pi}(t)\| \qquad (40) \\ \leq\ & D(\boldsymbol{\mu}(t)) - \left(\frac{1}{\gamma} - K_2\right)\|\boldsymbol{\pi}(t)\|^2 + K_1 \sum_{t'=t-2T}^{t-1} \|\boldsymbol{\pi}(t)\|^2 \qquad (41) \end{aligned}$$

Summing (40) over all t, we have

$$\begin{aligned} & D(\boldsymbol{\mu}(t+1)) \\ \leq\ & D(\boldsymbol{\mu}(0)) - \left(\frac{1}{\gamma} - K_2\right) \sum_{\tau=0}^{t} \|\boldsymbol{\pi}(\tau)\|^2 + K_1 \sum_{\tau=0}^{t} \sum_{t'=\tau-2T}^{\tau} \|\boldsymbol{\pi}(\tau)\|^2 \\ \leq\ & D(\boldsymbol{\mu}(0)) - \left(\frac{1}{\gamma} - K_2 - (2T+1)K_1\right) \cdot \sum_{\tau=0}^{t} \|\boldsymbol{\pi}(\tau)\|^2 \qquad (42) \end{aligned}$$

Choose γ sufficiently small such that $\frac{1}{\gamma} - K_2 - (2T+1)K_1 > 0$. Since $D(\boldsymbol{\mu}(t))$ is lower bounded, letting $t \to \infty$, we must have $\sum_{t=0}^{\infty} \|\boldsymbol{\pi}(t)\|^2 < \infty$, and hence

$$\|\boldsymbol{\pi}(t)\| \to 0 \text{ as } t \to \infty \qquad (43)$$

■

Summarizing above results, we establish **Theorem 2**.

Proof: We first prove that the various errors due to asynchronism all converge to zero. First

$$\begin{aligned} |\hat{\lambda}_f(t) - \lambda_f(t)| &= \sum_{t'=t-T}^{t} \Big| \sum_{l \in \mathcal{L}(f)} \rho_{lf}^{\alpha}(t',t)\mu_l^{\alpha}(t') - \mu_l^{\alpha}(t) + \sum_{f \to f'} \rho_{f'f}^{\beta}(t',t)\mu_{f'}^{\beta}(t') - \mu_{f'}^{\beta}(t) \Big| \\ &\leq \max_{t-T \leq t' \leq t} \Big| \sum_{l \in \mathcal{L}(f)} \mu_l^{\alpha}(t') - \mu_l^{\alpha}(t) + \sum_{f \to f'} \mu_{f'}^{\beta}(t') - \mu_{f'}^{\beta}(t) \Big| \\ &\leq \sum_{t'=t-T}^{t} \left(\sum_{l \in \mathcal{L}(f)} |\pi_l(t')| + \sum_{f \to f'} |\pi_{f'}(t')| \right) \\ &\leq \sum_{t'=t-T}^{t} \|\boldsymbol{\pi}(t')\|_1 \end{aligned}$$

which by (43) converges to zero as $t \to \infty$. Because $x_f(t)$ and $\overline{x}_f(t)$ are projections of $U_f^{'-1}$ onto $[m_f, M_f]$

and projection is nonexpansive [12] (Proposition 2.1.3), we have

$$\begin{aligned} |x_f(t) - \overline{x}_f(t)| &\leq |U_f'^{-1}(\hat{\lambda}_f(t)) - U_f'^{-1}(\lambda_f(t))| \\ &\leq \kappa_f \sum_{t'=t-T}^{t-1} \|\boldsymbol{\pi}(t')\|_1 \end{aligned}$$

Hence, by Eq. (43), $|x_f(t) - \overline{x}_f(t)| \to 0$ for all f.

We now show that every limit point of the sequence $\{\boldsymbol{\mu}(t)\}$ generated by the asynchronous algorithm minimizes the dual problem. Let $\boldsymbol{\mu}^*$ be a limit point of $\{\boldsymbol{\mu}(t)\}$. At least one exists, as it is constrained to lie in a compact set, provided γ is sufficiently small. Moreover, since the interval between consecutive updates is bounded (assumption **A4**), it follows that there exists a sequence of elements of T along which $\boldsymbol{\mu}$ converges. Let $\{t_k\}$ be a subsequence such that $\{\boldsymbol{\mu}(t_k)\}$ converges to $\boldsymbol{\mu}^*$. By **Lemma 5**, we have

$$\lim_k \boldsymbol{\xi}(t_k) = \lim_k \nabla D(\boldsymbol{\mu}(t_k)) = \nabla D(\boldsymbol{\mu}^*)$$

Hence

$$[\boldsymbol{\mu}^* - \gamma \nabla D(\boldsymbol{\mu}^*)]^+ - \boldsymbol{\mu}^* = \lim_k [\boldsymbol{\mu}(t_k) - \gamma \boldsymbol{\xi}(t_k)]^+ - \boldsymbol{\mu}(t_k) = \lim_k \boldsymbol{\pi}(t_k) = 0$$

Then by the projection theorem [12] (Proposition 2.1.3) and [11] (Proposition 3.3 in Section 3.3), we have that $\boldsymbol{\mu}^*$ minimizes D over $\boldsymbol{\mu} \geq 0$. By duality $\boldsymbol{x}^* = \boldsymbol{x}(\boldsymbol{\mu}^*)$ is the unique primal optimal rate. We now show that it is a limit point of $\{\boldsymbol{x}(t)\}$ generated by asynchronous algorithm. Consider a subsequence $\{\boldsymbol{x}(t_m)\}$ of $\{\boldsymbol{x}(t_k)\}$ such that $\{\boldsymbol{x}(t_m)\}$ converges. Since $\|\boldsymbol{x}(t) - \overline{\boldsymbol{x}}(t)\| \to 0$, we have

$$\lim_m \boldsymbol{x}(t_m) = \lim_m \overline{\boldsymbol{x}}(t_m) = \lim_m \boldsymbol{x}(\boldsymbol{\mu}(t_m)) = \boldsymbol{x}(\boldsymbol{\mu}^*)$$

which completes the proof. ■