Statistical Multiplexing over DSL Networks

Jianwei Huang[1], Chee Wei Tan[1], Mung Chiang[1], Raphael Cendrillon[2]
[1] Dept. of Electrical Engineering, Princeton University, NJ 08544, USA [2] Marvell Semiconductors Inc.
{jianweih,cheetan,chiangm}@princeton.edu, raphael@cendrillon.org

***Abstract*— Most previous work in statistical multiplexing only considered the case where the link transmission rates are fixed. In this paper, we consider statistical multiplexing in networks with *adaptive* transmission rates, with focus on DSL broadband access networks. This requires a jointly optimized allocation of buffer space and transmission bandwidth to traffic flows, which takes the flow traffic characteristics, the user QoS requirements, and the *user interactions at the physical layer* into consideration. Using the Effective Bandwidth concept, we propose a class of Alternate Maximization (AM) algorithms (AM-D and AM-M), which solve the statistical multiplexing problem for both delay insensitive Data traffic and delay sensitive Multimedia traffic. With low complexity as a design goal, the AM algorithms incorporate our recently proposed Autonomous Spectrum Balancing (ASB) algorithm, which was originally designed for DSL physical layer spectrum management. Our numerical results show that the AM algorithms combines the gain due to statistical multiplexing and that due to spectrum management.**

I. Introduction

A. Motivation

It is well known that provisioning bandwidth at the deterministic peak rates to bursty Internet traffic flows can guarantee packet lossless transmission. However, this approach underutilizes allocated resources during low activity periods [1]. An alternative approach is to provide *statistical service* that meets probabilistic Quality of Service (QoS) requirements [2], e.g., upperbound the packet loss probability:

$$\Pr\{\text{Packet Loss}\} < \epsilon \ , \tag{1}$$

where ϵ is a very small number (e.g., 10^{-6}). Allowing multiple bursty traffic flows to share a resource (e.g., bandwidth or buffer size), statistical properties among the flows can be exploited for efficient resource allocation. Statistical multiplexing is a technique that satisfies probabilistic QoS requirement such as (1) by allocating a bandwidth between its average and peak rate to each flow. With a fixed total bandwidth, the number of flows that can be supported under statistical service is larger compared to that under peak rate allocation. This performance gain is called the *statistical multiplexing gain*.

Resource allocation using statistical multiplexing has been widely studied in wireline networks (e.g., [1]–[6]) where the underlying link rate (e.g., transmission bandwidth) is assumed to be fixed, and also in the wireless setting (e.g., [7]–[10]) where wireless channel variation (leading to changing bandwidth allocation) is considered. Bandwidth allocation at the link layer depends on the achievable rate at the physical layer. Thus, the link and physical layer resource allocation can be jointly optimized. No previous work in wireline networks has considered a joint optimization of the physical layer achievable rate and resource allocation (e.g., bandwidth, buffer) at the link layer with statistical multiplexing.

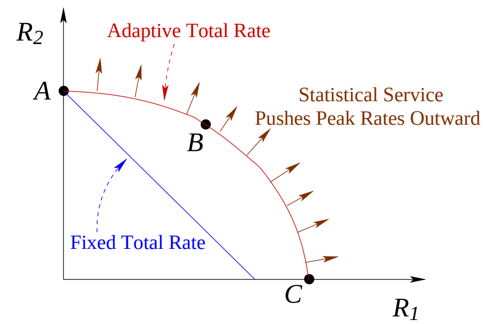


Fig. 1. Rate regions for both fixed and variable total rate cases. The *statistical service* pushes the peak rate pair (R_1, R_2) outside the rate region.

Fig. 1 illustrates the following two different choices in rate allocation using a two-user example: (i) deterministic versus statistical services, (ii) fixed versus adaptive total rate summed over two users. The horizontal (vertical) axis represents the rate achieved by user 1 (user 2). A point in the positive quadrant is characterized by a *peak rate pair* (R_1, R_2). The fixed total rate case corresponds to a simplex rate region, i.e., $R_1 \geq 0$, $R_2 \geq 0$, and $R_1 + R_2 \leq R^{\max}$ (where $R^{\max}$ is the fixed total rate). The adaptive total rate corresponds to a rate region with a nonlinear boundary (i.e., $A \to B \to C$). For example, point B gives a larger total peak rate $(R_1 + R_2)$ than either point A or point C. The two rate regions plotted here both correspond to the deterministic service case where the peak rate pair (R_1, R_2) must be on or within the rate region boundary. In the statistical service case, the peak rate pair can be pushed *outside* of the achievable rate region (shown by the arrows) since the system is designed to tolerate a given probabilistic loss.[1]

In Digital Subscriber Line (DSL) networks, achievable rates in the physical layer may be adversely affected by strong crosstalk interferences. However, the achievable rate of each line can be adaptively changed by regulating the transmit power of all users sharing the same binder. This is known as *dynamic spectrum management*. The attained interference-limited rate region is in general nonconvex. It has however been shown to be convex assuming an asymptotically large number of frequency tones [11]. Tunable network performance can be obtained by picking a point at the Pareto boundary of the rate region.

[1]Note that the "rate region" used here is defined on the achievable *peak* rates of the users, instead of the average arrival rates as in many other literature.

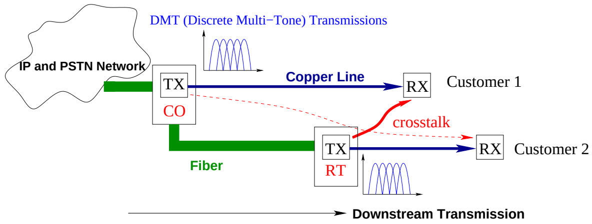


Fig. 2. Downstream transmission in a two-line DSL network. The CO is connected to the IP and PSTN Network via fiber link; the RT is connected to the CO via fiber as well. The CO and the RT terminate at end customer homes through copper twisted-pair lines (telephone lines), where data rate is limited by crosstalk.

B. Background on DSL System Model

Fig. 2 shows a DSL system model of two copper lines (users). The first line is from the Central Office (CO) to customer 1. Since customer 2 is located far away from CO, the service provider deploys a Remote Terminal (RT) near the edge of the network, which connects with customer 2 through a relatively short copper line. The CO is connected to the PSTN/IP network through a fiber link, and the RT is connected to the CO through a fiber link as well.[2]

In the downstream transmission case shown in the figure, the transmitting modems are located at the CO and RT, and the receivers are at the customer homes. Each DSL modem transmits over multiple frequency tones. Multiple lines sharing the same binder generate crosstalks (interferences) to each other on all frequency tones. In the mixed CO/RT case, the RT generates excessive interference to the CO line due to the physical proximity between the RT transmitter and the receiver on the CO line.

Recent work has improved DSL performance by viewing the shared binder of twisted pairs as one aggregate multi-user communication system, where the transmit power (spectrum) of each user at the physical layer can be dynamically adjusted to achieve a rate region that is much larger than that in practice. Different spectrum management algorithms (e.g., [11]–[15]) lead to rate regions with different shapes. The ASB algorithm proposed in [15] is the first algorithm that is both fully distributed and attain near-optimal rate region for the multi-carrier interference DSL channel, and is used to design the algorithms proposed in this paper.

The results in this paper are obtained as part of the *FAST Copper* project, www.princeton.edu/fastcopper, which is a joint project between Princeton University, Stanford University, and Fraser Research Lab, under the sponsorship of U.S. National Science Foundation and in collaboration with AT&T broadband access network deployment. The goal is to provide an order-of-magnitude increase in DSL broadband access speed through a joint optimization of resources in Frequency, Amplitude, Space, and Time (thus "FAST"), so that economically viable broadband access can be ubiquitously deployed in U.S. Exploiting traffic burstiness over a spectrum-managed interference channel is part of the Time dimension considered in the project.

[2]In practice, there are multiple lines connecting the same CO (or RT) to multiple customer homes. For simplicity, we only focus on the two-line model.

C. Summary of Contributions

Our contributions in this paper are as follows:

- *Framework*: The use of statistical multiplexing concept over spectrum-managed broadband access networks is illustrated using an optimization formulation that considers the link layer network resources (e.g., bandwidth, buffer) and achievable rates at the physical layer.
- *Algorithm:* We propose Alternate Maximization (AM) algorithms, which are autonomous and have low-complexity. The algorithms are designed for both data and multimedia traffic. The AM algorithms decompose the statistical multiplexing problem into a bandwidth allocation stage and a buffer allocation stage. The two stages are solved iteratively to improve the system performance.
- *Performance:* Our algorithms outperform systems that do not utilize either spectrum management or statistical multiplexing. Using a DSL network simulator, we show that our algorithm can admit as much as 200% of the total flows of a system without spectrum management. The statistical multiplexing gain achieved is up to 180% in a two-line DSL network, which provides a useful benchmark for the design of DSL networks.

The rest of the paper is organized as follows. In Sect. II, we describe the DSL network model with both downstream and upstream transmissions, as well as the main problem formulation. In Sect. III, we propose the AM-D algorithm to solve the statistical multiplexing problem with data traffic only. In Sect. III, we extend our result to mixed data and multimedia traffic using the AM-M algorithm. An Iterative Hypothesis Testing (IHT) algorithm is proposed to solve the bandwidth allocation stage of the AM-M algorithm. Then, we present several numerical examples in Sect. V to illustrate the performance gains of the proposed algorithms. Conclusions are given in Sect. VI and the ASB algorithm is briefly described in the Appendix.

II. Network Model and Problem Formulation

We consider the downstream and upstream DSL network models as shown in Fig. 3. In both cases, we have N DSL users (copper wire links) that share a common multiplexing link. In the downstream case, the multiplexing link (which could be a fiber link) has total buffer size B that is shared by a set $\mathcal{N} = \{1, ..., N\}$ users, indexed by i. Each user can have several classes of applications, indexed by j. In the upstream case, user i has fixed buffer space B_i located at its individual link. The achievable rates of the DSL links are determined by crosstalks among users, which are in turn determined by modems' transmit power and the channel gains. The maximum achievable rate over the multiplexing link depends on the total rates across all DSL links. The QoS requirements of the downstream and upstream applications depend on the bandwidth and buffer allocations. In the downstream case, we need to optimize the achievable rates of all DSL links as well as the buffer partition at the multiplexing link. The upstream case is similar to the downstream case except that there is no

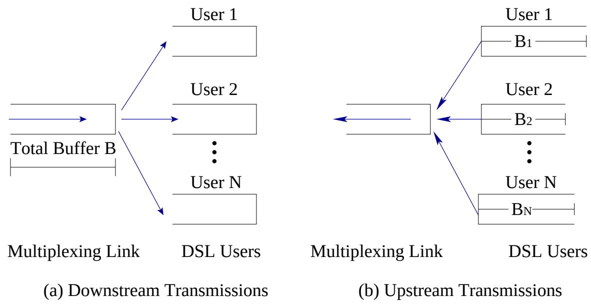


Fig. 3. Downstream and upstream transmission schemes.

need to consider buffer allocation.[3]

We divide the scheduling time axis into "statistical multiplexing intervals", and each interval is sufficiently long to exploit the stationary stochastic nature of the traffic. At the beginning of each interval, we want to solve Problem (2) below. In Sects III and IV, we discuss special cases of Problem (2). For user i and application j, we use w_i^j to denote the weight coefficient, a_i^j to denote the average rate, n_i^j to denote the number of admitted flows, B_i^j to denote the allocated buffer space, ϵ_i^j to denote the QoS parameter, q_i^j to denote the QoS function (function of the buffer allocation B_i^j and the bandwidth allocations c_i/n_i^j, where c_i is the total bandwidth allocated to user i). The constraints include the QoS constraint $q_i^j\left(B_i^j, \frac{c_i}{n_i^j}\right) \le \epsilon_i^j$ (specific form of function q_i^j are given in later sections), the total buffer space constraint, and the rate constraint $\boldsymbol{c} \in \mathcal{C}$ (where $\mathcal{C}$ denotes the *maximum* achievable rate region of the DSL network). We want to determine the buffer allocation $\boldsymbol{B} = \left(B_i^j, \forall i, j\right)$, the bandwidth allocation $\boldsymbol{c} = (c_i, \forall i)$, and the number of flows admitted to the system $\boldsymbol{n} = \left(n_i^j, \forall i, j\right)$. Consider the following problem formulation:

$$\begin{aligned} \text{maximize} \quad & \sum_i \sum_j w_i^j a_i^j n_i^j && (2) \\ \text{subject to} \quad & q_i^j\left(B_i^j, \frac{c_i}{n_i^j}\right) \le \epsilon_i^j, \forall i, j \\ & \sum_i \sum_j B_i^j = B \quad \text{(downstream only)} \\ & \sum_j B_i^j = B_i, \forall i \quad \text{(upstream only)} \\ & \boldsymbol{c} \in \mathcal{C} \\ \text{variables} \quad & \boldsymbol{n}, \boldsymbol{B}, \boldsymbol{c} \ge \boldsymbol{0} \end{aligned}$$

The weights w_i^j's are fixed in Problem (2), but can be adjusted between statistical multiplexing intervals. They may be regarded as functions of users' achieved throughput or instantaneous queue lengths.

Key notation used in this paper is summarized in Table I. We use bold symbols to denote vectors and superscript $(*)$ to denote optimal solutions.

[3]In this paper, we use the terms *rate* and *bandwidth* interchangeably to denote the peak rate a link can provide.

TABLE I

SUMMARY OF KEY NOTATION USED IN THE PAPER

Symbol	Meaning
i	user index
j	application index (d: data; m: multimedia)
k	multimedia state space index
N	total number of users
K	size of state space of multimedia traffic
w_i^j	weight coefficient of application j of user i
n_i^j	number of flows of application j of user i
a_i^j	average rate of a flow of application j of user i
r_i^j	Peak rate of a flow of application j of user i
B_i^j	buffer allocated to a flow of application j of user i
B_i	total buffer allocated to user i
ϵ_i^j	QoS parameter of application j of user i
$q_i^j(\cdot)$	QoS function of application j of user i
$g_i(\cdot)$	Effective bandwidth of one data flow of user i
δ_i^j	Spatial parameter of application j of user i
c_i	total rate/bandwidth allocated to user i
$\mathcal{C}$	maximum rate region of the DSL network
f_i^j	flow of application j of user i
$\boldsymbol{h}_i$	user i's multimedia flow state space
$\boldsymbol{p}_i$	user i's multimedia flow marginal distribution
$M_i(\theta)$	Moment generating function of user i's multimedia traffic
$\overline{w}_i$	user i's weight in the ASB algorithm
bold symbol	vectors
superscript $*$	optimal values

III. Multiplexing Delay Insensitive Data Traffic

In this section, we assume each user i only has a single application class, which is delay insensitive data traffic (denoted by a superscript d). We first discuss how to characterize the "average" resource consumption of such traffic using the concept of effective bandwidth (discussed in many papers, e.g., [16], [17]). We then propose an AM-D algorithm to solve Problem (2).

A. *Effective Bandwidth of Delay Insensitive Data Traffic*

Assume that the user i's data flow can be modeled as a Levy process $f_i^d(t)$ (i.e., process with stationary independent increments [17]), which has an average rate a_i^d and a peak rate r_i^d. We can approximate the process $f_i^d(t)$ as a *constant rate* traffic with rate $g_i^d\left(\delta_i^d\right)$, known as the *effective bandwidth*. δ_i^d is called the spatial parameter, and $g_i(\cdot)$ is an increasing function in δ_i^d: $g_i\left(\delta_i^d\right)$ approaches a_i^d when $\delta_i^d \to 0$, and approaches r_i^d when δ_i^d becomes large. δ_i^d depends on the QoS requirements as well as the traffic characteristics of $f_i^d(t)$.

The specific form of $g_i(\cdot)$ depends on the statistical property of $f_i^d(t)$. For example, consider a compound Poisson arrival process with a total arrival data volume in a time interval $[0, t]$ as

$$X[0, t] = \sum_{n=1}^{A(t)} Y_n,$$

where $Y_1, Y_2, ..., Y_{A(t)}$ are independent identically distributed random variables with distribution F, and $A(t)$ is an independent Poisson process of rate λ, the effective bandwidth is given by [16], [17]:

$$g(\delta) = \frac{1}{\delta} \int \left(e^{\delta x} - 1\right) \lambda \, dF(x).$$

In the case where $Y_1, Y_2, ..., Y_{A(t)}$ are exponentially distributed with parameter μ, then

$$g(\delta) = \frac{\lambda}{\mu - \delta} \quad \text{for} \quad \delta < \mu.$$

The average rate of the flow equals λ/μ.

We define the QoS parameter of user i's data traffic as the probability of packet loss (or buffer overflow), ϵ_i^d. It has been shown in [17] that an upperbound on this probability decays exponentially fast with the allocated buffer B_i when the buffer size is large, and satisfies the following relationship

$$\Pr\{\text{Packet Loss}\} \leq e^{-B_i \delta_i^d}.$$

By letting the upperbound equal to ϵ_i^d, we have

$$\delta_i^d = -\frac{\log \epsilon_i^d}{B_i} > 0,$$

which increases when the QoS requirement becomes more stringent (ϵ_i^d decreases) or B_i decreases, and means more bandwidth is required to achieve the QoS target. The first constraint (QoS constraint) in Problem (2) is then replaced by $g_i\left(-\frac{\log \epsilon_i^d}{B_i}\right) \leq c_i/n_i$, i.e., the actual bandwidth allocated to each data flow should be no smaller than its effective bandwidth for given B_i and ϵ_i^d.

B. Bandwidth Allocation for the Upstream Case

We first consider the upstream transmission case (see Fig. 3), where each DSL user has a separate fixed buffer space. We only need to determine the bandwidth of each DSL link and the number of flows admitted on each link, subject to the fixed QoS constraints ϵ_i. Since B_i and ϵ_i are fixed, the effective bandwidth $g_i\left(-\frac{\log \epsilon_i^d}{B_i}\right)$ is fixed, and can be denoted by $\overline{g}_i \stackrel{\text{def}}{=} g_i\left(-\frac{\log \epsilon_i^d}{B_i}\right)$. The problem formulation simplifies to

$$\begin{aligned} \text{maximize} \quad & \sum_i w_i^d a_i^d n_i^d \\ \text{subject to} \quad & \overline{g}_i \leq \frac{c_i}{n_i^d}, \forall i, \\ & \boldsymbol{c} \in \mathcal{C} \\ \text{variables} \quad & \boldsymbol{n}^d, \boldsymbol{c} \geq \boldsymbol{0} \end{aligned} \tag{3}$$

Intuitively, we admit more flows for user i if it has a higher weight w_i^d, its flow has a higher average rate a_i^d, and a lower effective bandwidth requirement $\overline{g}_i$ (which implies a less stringent QoS requirement ϵ_i^d or a larger B_i). Before solving Problem (3), we make the following two observations:

Observation 1: The stochastic nature of the data traffic has been captured by the effective bandwidth function $\overline{g}_i$, so Problem (3) is a deterministic optimization formulation.

Observation 2: At optimality, the QoS constraint is tight, i.e., $c_i = \overline{g}_i n_i^d$, so $\boldsymbol{n}^d$ and $\boldsymbol{c}$ are dependent on each other.

These two observations enable us to further simplify Problem (3). Define $c_i = n_i^d \overline{g}_i$ and a new weight coefficient $\overline{w}_i = w_i^d a_i^d / \overline{g}_i$. Problem (3) can be written as

$$\begin{aligned} \text{maximize} \quad & \sum_i \overline{w}_i c_i \\ \text{subject to} \quad & \boldsymbol{c} \in \mathcal{C}, \\ \text{variables} \quad & \boldsymbol{c} \geq \boldsymbol{0} \end{aligned} \tag{4}$$

This becomes a standard weighted rate maximization problem subject to achievable rate constraint, and can be solved using the ASB algorithm (see Appendix) distributively to determine the bandwidth allocation $\boldsymbol{c}$ and the number of admitted flows $\boldsymbol{n}^d = \boldsymbol{c}/\overline{\boldsymbol{g}}$.

C. Capacity and Buffer Allocation for the Downstream Case

Next, we consider the downstream case where we also need to consider the buffer partition problem across users (i.e., determining the buffer vector $\boldsymbol{B} = (B_1, B_2, ..., B_N)$. The problem we want to solve is given as follows:

$$\begin{aligned} \text{maximize} \quad & \sum_i w_i^d a_i^d n_i^d \\ \text{subject to} \quad & g_i\left(-\frac{\log \epsilon_i^d}{B_i}\right) \leq \frac{c_i}{n_i^d}, \forall i, \\ & \sum_i B_i = B. \\ & \boldsymbol{c} \in \mathcal{C} \\ \text{variables} \quad & \boldsymbol{n}^d, \boldsymbol{c}, \boldsymbol{B} \geq \boldsymbol{0} \end{aligned} \tag{5}$$

Here, the effective bandwidth of user i's data flow $g_i\left(-\frac{\log \epsilon_i^d}{B_i}\right)$ is no longer a constant, since it depends on the variable B_i. In general, Problem (5) is non-convex (due to the multiplicative form of the QoS constraint), making it hard to solve for the globally optimal solution.

We propose an *Alternate Maximization algorithm for Data traffic (**AM-D** algorithm)* to reach a suboptimal solution of Problem (5). The AM-D algorithm consists of two stages: bandwidth allocation stage and buffer allocation stage. During each stage, we fix some of the optimization variables and solve for the rest, which is easier to do than solving for all variables simultaneously. The AM-D algorithm iterates through the two stages described as follows until all variables converge. The two stages are given as:

1) Stage 1 (bandwidth allocation, i.e., fix $\boldsymbol{B}$ and solve for $(\boldsymbol{n}^d, \boldsymbol{c})$): In this stage, we assume that the buffer allocations to all users ($\boldsymbol{B}$) are fixed. Problem (5) thus reduces to the upstream capacity allocation problem as in Problem (3).

2) Stage 2 (buffer allocation, i.e., fix $\boldsymbol{c}$ and solve for $(\boldsymbol{n}^d, \boldsymbol{B})$): At this stage, we assume that the capacity allocation $\boldsymbol{c}$ (for all users) is fixed. We want to find the optimal buffer allocation $\boldsymbol{B}$ and the number of admitted flows $\boldsymbol{n}^d$. For a given user i, the more buffer we allocate, the smaller the effective bandwidth $g_i\left(-\frac{\log \epsilon_i^d}{B_i}\right)$ is, and the more number of flows (n_i^d) that can be admitted. The problem that we want to solve is given by

$$\begin{aligned} \text{maximize} \quad & \sum_i w_i^d a_i^d n_i^d \qquad (6) \\ \text{subject to} \quad & g_i\left(-\frac{\log \epsilon_i^d}{B_i}\right) \le \frac{c_i}{n_i^d}, \forall i, \\ & \sum_i B_i = B. \\ \text{variables} \quad & \boldsymbol{n}^d, \boldsymbol{B} \ge \boldsymbol{0} \end{aligned}$$

Observation 3: At the optimal solution of Problem (6), the QoS constraint is tight, i.e., $n_i^d g_i\left(-\frac{\log \epsilon_i^d}{B_i}\right) = c_i$.

To simplify problem (6), we define a function $y_i\left(B_i\right) \stackrel{\text{def}}{=} c_i / g_i\left(-\frac{\log \epsilon_i^d}{B_i}\right)$, which is increasing in B_i. Problem (6) can thus be rewritten as

$$\begin{aligned} \text{maximize} \quad & \sum_i w_i^d a_i^d y_i\left(B_i\right) \qquad (7) \\ \text{subject to} \quad & \sum_i B_i = B, \\ \text{variables} \quad & \boldsymbol{B} \ge \boldsymbol{0} \end{aligned}$$

Since any increasing function is quasi-concave, Problem (7) is a quasi-concave maximization problem, which can be solved using bisection search (by solving a sequence of feasibility problems, see [18]). However, for special cases of $y_i(B_i)$ (e.g., compound Poisson traffic), Problem (7) is a strictly concave maximization problem, which has a closed form solution as given in the following result.

Proposition 1: If data flows follow compound Poisson arrival distribution with exponentially distributed file size, i.e., $g_i(\delta_i) = \frac{\lambda_i}{\mu_i - \delta_i}$, for $\delta_i < \mu_i$, the optimal solution to Problem (7) is given by

$$B_i^* = \frac{\sqrt{w_i^d c_i |\log \epsilon_i^d| / \mu_i}}{\sum_j \sqrt{w_j^d c_j |\log \epsilon_i^d| / \mu_j}} B \quad \forall\ i. \qquad (8)$$

The AM-D algorithm iterates through the two stages until the variables $\left(\boldsymbol{n}^d, \boldsymbol{c}, \boldsymbol{B}\right)$ converge to a solution. The complete algorithm is given in Algorithm 1:

Algorithm 1 AM-D: Alternate Maximization for Data Traffic

1: **repeat**
2: Determine $\left(\boldsymbol{n}^d, \boldsymbol{c}\right)$ under fixed $\boldsymbol{B}$: solve Problem (4) using ASB algorithm.
3: Determine $\left(\boldsymbol{n}^d, \boldsymbol{B}\right)$ under fixed $\boldsymbol{c}$: solve Problem (7) using bi-section search (or in closed form as in (8) for compound Poisson traffic).
4: **until** Convergence

Since each stage of the AM-D algorithm improves the objective of Problem (5), which is upperbounded, we have the following result.

Theorem 1: The AM-D algorithm converges to a feasible solution of Problem (5).

Although we cannot verify the suboptimality gap between the solution obtained by the AM-D solution and the globally optimal solution, our numerical tests in Sect. V show that the AM-D algorithm improves the performance as compared to the case without considering statistical multiplexing. Since upstream transmission is a special case of downstream transmission (i.e., no buffer allocation in upstream), the AM-D algorithm is also applicable to the upstream case.

IV. Multiplexing Data and Multimedia Traffic

In the previous section, we simplify the statistical multiplexing problem for delay insensitive data flows using the concept of effective bandwidth. In this section, we will further consider multimedia flows which typically have stringent delay requirements. This delay sensitivity precludes a straightforward application of the effective bandwidth concept, where a large buffer size implies a large delay. It has been shown in [19] that many multimedia applications (e.g., smoothed video traffic) can only be supported successfully (e.g., satisfy the delay and loss probability requirements) by allocating enough bandwidth to the flows. Motivated by this, we will assume a bufferless model for multimedia flows. Specifically, we follow an approach similar to [19], which calculates the bandwidth requirements and loss probabilities of multimedia traffic based on the multimedia flows' marginal distributions. We assume that if the instantaneous multimedia traffic rate exceeds the allocated bandwidth, the arriving packets are immediately dropped instead of being stored in a buffer.

In this section, we consider the statistical multiplexing problem of a mixed data and multimedia traffic. We use superscript d to represent data flows and superscript m to represent multimedia flows.

A. Multimedia QoS Requirement based on the Chernoff Bound

Assume that user i's multimedia flow can be modeled as a stationary random process $f_i^m(t)$, which has a marginal distribution described by states $\boldsymbol{h}_i = (h_{i,1}, ..., h_{i,K_i})$ and associated probability $\Pr\{f_i^m(t) = h_{i,k}\} = p_{i,k}$ (independent of time t). Assuming different multimedia flows of the same user are mutually independent, the loss probability of user i can be approximated by the Chernoff bound [19]:

$$\Pr\{\text{user } i\text{'s multimedia traffic volume} \ge c_i^m\} \approx e^{-\Lambda_i^*(c_i^m)},$$

where c_i^m is the bandwidth allocated to user i's multimedia traffic, and

$$\Lambda_i^*\left(\mu\right) \stackrel{\text{def}}{=} \sup_{\theta \ge 0} \left\{\theta\mu - \Lambda_i\left(\theta\right)\right\}, \qquad (9)$$

$$\Lambda_i\left(\theta\right) \stackrel{\text{def}}{=} n_i^m \log M_i\left(\theta\right), \qquad (10)$$

$$M_i\left(\theta\right) \stackrel{\text{def}}{=} \sum_{k=1}^{K_i} p_{i,k} e^{\theta h_{i,k}}, \qquad (11)$$

where $M_i\left(\theta\right)$ is the moment generating function.

The Chernoff bound can be used to estimate the minimum bandwidth c_i^m needed to achieve a packet loss probability ϵ_i^m in a bufferless model, i.e.,

$$\Pr\{\text{user } i\text{'s total multimedia traffic volume} \ge c_i^m\} \le \epsilon_i^m.$$

To meet the QoS requirement, we want

$$e^{-\Lambda_i^*(c_i^m)} = \epsilon_i^m.$$

After some manipulation, the minimum bandwidth needed by user i's multimedia traffic can be calculated by

$$c_i^m = n_i^m \frac{M_i'\left(\theta_i^*\right)}{M_i\left(\theta_i^*\right)}, \tag{12}$$

where θ_i^* is the optimal solution to (9) and satisfies the following equation:

$$\theta_i^* \Lambda_i'\left(\theta_i^*\right) - \Lambda_i\left(\theta_i^*\right) = \left|\log \epsilon_i^m\right|. \tag{13}$$

Here, $M_i'\left(\theta_i\right) = \partial M_i\left(\theta_i\right)/\partial\theta_i$ and $\Lambda_i'\left(\theta_i\right) = \partial\Lambda_i\left(\theta_i\right)/\partial\theta_i$. Substitute the definition of $\Lambda_i\left(\theta\right)$ in (10) into (13), and define

$$q_i\left(\theta_i\right) \stackrel{\text{def}}{=} \theta_i \frac{M_i'\left(\theta_i\right)}{M_i\left(\theta_i\right)} - \log M_i\left(\theta_i\right), \tag{14}$$

we obtain the relationship between the number of multimedia flows n_i^m and the QoS requirement ϵ_i^m,

$$n_i^m q_i\left(\theta_i^*\right) = \left|\log \epsilon_i^m\right|. \tag{15}$$

Using (12) and (15), we can determine the values of n_i^m and θ_i^* (as functions of c_i^m and ϵ_i^m).

Since the upstream problem is a special case of the downstream problem (i.e., without buffer allocation), we only consider the downstream case next.

B. Bandwidth and Buffer Allocation for the Downstream Case

We want to solve the following problem:

$$\begin{aligned}
\text{maximize} \quad & \sum_i \left(w_i^d a_i^d n_i^d + w_i^m a_i^m n_i^m\right) \qquad (16)\\
\text{subject to} \quad & g_i\left(-\frac{\log \epsilon_i^d}{B_i}\right) \le \frac{c_i^d}{n_i^d}, \forall i,\\
& n_i^m q_i\left(\theta_i\right) = \left|\log \epsilon_i^m\right| \mathbf{1}_{\left\{c_i^m>0\right\}}, \forall i,\\
& n_i^m \frac{M_i'\left(\theta_i\right)}{M_i\left(\theta_i\right)} = c_i^m, \forall i,\\
& c_i^d + c_i^m = c_i, \forall i,\\
& \sum_i B_i = B,\\
& \boldsymbol{c} \in \mathcal{C}.\\
\text{variables} \quad & \boldsymbol{n}^d, \boldsymbol{n}^m, \boldsymbol{c}^d, \boldsymbol{c}^m, \boldsymbol{c}, \boldsymbol{B}, \boldsymbol{\theta} \ge \mathbf{0}
\end{aligned}$$

The first constraint corresponds to the QoS requirements of data flows. The second and third constraints correspond to the QoS requirements of multimedia flows obtained from Chernoff bound. The indicator function $\mathbf{1}_{\{c_i^m>0\}}$ is necessary since the second constraint is only meaningful if positive bandwidth is allocated to a user's multimedia traffic. The fourth constraint means the bandwidth allocated to any user is split between data and multimedia traffic (though one of the applications might get zero bandwidth). The last two constraints are for downstream buffer space and maximum achievable rate region. By solving Problem (16), we will determine the number of data and multimedia flows $\left(\boldsymbol{n}^d, \boldsymbol{n}^m\right)$, the bandwidth allocation $\left(\boldsymbol{c}^d, \boldsymbol{c}^m, \boldsymbol{c}\right)$, and the buffer allocation $\left(\boldsymbol{B}\right)$. The vector variable $\boldsymbol{\theta}$ is auxiliary, and is used when applying Chernoff bound to estimate the performance of the multimedia traffic.

Problem (16) is nonconvex, and thus is difficult to solve for the global optimum. We propose an *Alternate Maximization algorithm for data and Multimedia traffic (**AM-M** algorithm)* to find a local optimal solution of Problem (16). The AM-M algorithm takes a similar iterative approach as the AM-D algorithm in Sect. III-C. The key difference between the AM-M and AM-D algorithms lies in the bandwidth allocation stage, where we determine how to share common resources between data and multimedia flows. The buffer allocation stage, on the other hand, is the same in both AM-M and AM-D algorithms because we use a bufferless model for multimedia flows, and we allocate buffers only for data flows. The two stages of AM-M algorithm are described as follows.

1) Stage 1 (bandwidth allocation, i.e., fix $\boldsymbol{B}$, solve $\left(\boldsymbol{n}^d, \boldsymbol{n}^m, \boldsymbol{c}^d, \boldsymbol{c}^m, \boldsymbol{c}, \boldsymbol{\theta}\right)$): We will solve Problem (16) using a series of ASB algorithms. Let us define

$$\begin{aligned}
\overline{w}_i^d &\stackrel{\text{def}}{=} w_i^d a_i^d / g_i\left(-\frac{\log \epsilon_i^d}{B_i}\right),\\
\overline{w}_i^m\left(\theta_i\right) &\stackrel{\text{def}}{=} w_i^m a_i^m M_i\left(\theta_i\right)/M_i'\left(\theta_i\right),\\
u_i\left(\theta_i\right) &\stackrel{\text{def}}{=} \theta_i - \log M_i\left(\theta_i\right) \frac{M_i\left(\theta_i\right)}{M_i'\left(\theta_i\right)}.
\end{aligned}$$

Note that $\overline{w}_i^d$ is a constant (due to fixed B_i in this stage), $\overline{w}_i^m\left(\theta_i\right)$ is decreasing in θ_i, and $u_i\left(\theta_i\right)$ is nonnegative, and increasing in θ_i.

With fixed $\boldsymbol{B}$, Problem (16) can be rewritten as:

$$\begin{aligned}
\text{maximize} \quad & \sum_i \left(\overline{w}_i^d c_i^d + \overline{w}_i^m\left(\theta_i\right) c_i^m\right) \qquad (17)\\
\text{subject to} \quad & c_i^m u_i\left(\theta_i\right) = \left|\log \epsilon_i^m\right| \mathbf{1}_{\left\{c_i^m>0\right\}}, \forall i,\\
& c_i^d + c_i^m = c_i, \forall i,\\
& \boldsymbol{c} \in \mathcal{C}.\\
\text{variables} \quad & \boldsymbol{c}^d, \boldsymbol{c}^m, \boldsymbol{c}, \boldsymbol{\theta} \ge \mathbf{0}
\end{aligned}$$

Assume that we know the optimal $\boldsymbol{\theta}^*$, then the objective function of Problem (17) becomes a linear function in rates $\left(\boldsymbol{c}^d, \boldsymbol{c}^m\right)$. This leads to the following result:

Lemma 1: At the optimal solution of Problem (17), one of the following is true for each user i:

1) $\overline{w}_i^d > \overline{w}_i^m\left(\theta_i^*\right)$, $c_i^{d*} = c_i^*$ and $c_i^{m*} = 0$.
2) $\overline{w}_i^d = \overline{w}_i^m\left(\theta_i^*\right)$, $c_i^{d*} + c_i^{m*} = c_i$.
3) $\overline{w}_i^d < \overline{w}_i^m\left(\theta_i^*\right)$, $c_i^{d*} = 0$ and $c_i^{m*} = c_i$.

Lemma 1 shows that all the bandwidth allocated to a user i is used to transmit the traffic that has a higher weight (if both data and multimedia traffic have same weights as in case 2, a user can split the bandwidth).

The implications for solving Problem (17) are as follows: If all users fall into either case 1 or case 2, then we can let $\overline{w}_i = \overline{w}_i^d$ and solve Problem (4) using the ASB algorithm. Then user i may either transmit data traffic only or split the bandwidth c_i between data and multimedia traffic, depending on whether it is case 1 or case 2. The main difficulty lies in case 3, where user i transmits multimedia traffic only. In this case, the bandwidth allocation depends on the value of $\overline{w}_i^m\left(\theta_i\right)$, which is a function of variable θ_i, and θ_i needs to be chosen such that the constraint $c_i^m u_i\left(\theta_i\right) = \left|\log \epsilon_i^m\right|$ is satisfied at the final solution. This precludes a direct application of the ASB algorithm which can only handle fixed weights.

In short, we need to answer the following two questions for *each user* i:

1) Is $\overline{w}_i^d \geq \overline{w}_i^m(\theta_i)$ at optimality?
2) If not, how large should θ_i be such that the constraint is met and the objective function of Problem (17) is maximized?

To answer the above two questions, we propose an *Iterative Hypothesis Testing (IHT) Algorithm* to solve Problem (17). The following notation is used. Denote $c_i(\theta_i, \overline{w}_{-i})$ as the bandwidth allocated to user i by solving Problem (4) using ASB algorithm with $\overline{w}_i = \overline{w}_i^m(\theta_i)$ and fixed $\overline{w}_{-i} = (\overline{w}_1, ..., \overline{w}_{i-1}, \overline{w}_{i+1}, ...\overline{w}_N)$.

The IHT algorithm is given in Algorithm 2. Lines 1 and 2 solve Problem (17) using ASB algorithm (temporarily assuming $\overline{w}_i^d \geq \overline{w}_i^m(\theta_i^*)$ is true for all users at the optimal solution). In line 5, user i uses the Hypothesis Testing (HT) subroutine (Algorithm 3) to see whether the objective of Problem (17) can be further improved by transmitting multimedia traffic only.

Algorithm 2 IHT: Iterative Hypothesis Testing Algorithm

1: $\overline{w}_i \Leftarrow \overline{w}_i^d$ for all $i \in \mathcal{N}$.
2: $\boldsymbol{c} \Leftarrow$solution of Problem (4).
3: **repeat**
4:     **for all** user $i \in \mathcal{N}$ **do**
5:         $(\overline{\boldsymbol{w}}, \boldsymbol{c}) \Leftarrow HT(i, \overline{\boldsymbol{w}}, \boldsymbol{c})$.
6:     **end for**
7: **until** Convergence

Algorithm 3 Hypothesis Testing (HT) subroutine

1: **procedure** HT($i, \overline{\boldsymbol{w}}, \boldsymbol{c}$)
2:     **if** $\overline{w}_i \leq \overline{w}_i^m\left(\tilde{\theta}_i^\infty\right)$ **then**
3:         $\tilde{\theta}_i \Leftarrow \tilde{\theta}_i^\infty$.
4:     **else if** $\overline{w}_i^m\left(\tilde{\theta}_i^\infty\right) < \overline{w}_i \leq \overline{w}_i^m(0)$ **then**
5:         $\tilde{\theta}_i \Leftarrow \arg_{\theta_i}(\overline{w}_i^m(\theta_i) = \overline{w}_i)$.
6:     **else**
7:         $\tilde{\theta}_i \Leftarrow 0$.
8:     **end if**
9:     **if** $\tilde{\theta}_i > 0$ **then**
10:         **if** $\exists \widehat{\theta}_i \leq \tilde{\theta}_i$, s.t. $c_i\left(\widehat{\theta}_i, \overline{w}_{-i}\right) u_i\left(\widehat{\theta}_i\right) = |\log \epsilon_i^m|$ **then**
11:             $\overline{w}_i \Leftarrow \overline{w}_i^m\left(\widehat{\theta}_i\right)$.
12:             $\boldsymbol{c} \Leftarrow$solution of Problem (4).
13:         **end if**
14:     **end if**
15:     **return** $(\overline{\boldsymbol{w}}, \boldsymbol{c})$
16: **end procedure**

In the HT subroutine, lines 2 to 8 find a threshold $\tilde{\theta}_i$ such that $\overline{w}_i^m(\theta_i) \geq \overline{w}_i$ for all $\theta_i \leq \tilde{\theta}_i$. Here $\tilde{\theta}_i^\infty$ is a large enough finite value such that $\overline{w}_i^m\left(\tilde{\theta}_i^\infty\right)$ is very close to the limiting value $\overline{w}_i^m(\infty)$. Choosing a finite $\tilde{\theta}_i^\infty$ avoids numerical instability in the computation. Line 10 in the HT subroutine checks if there exists a $\widehat{\theta}_i \leq \tilde{\theta}_i$ such that $c_i\left(\widehat{\theta}_i, \overline{w}_{-i}\right) u_i\left(\widehat{\theta}_i\right) = |\log \epsilon_i^m|$. However, this is in general difficult to verify since $c_i\left(\widehat{\theta}_i, \overline{w}_{-i}\right) u_i\left(\widehat{\theta}_i\right)$ does not have an analytical form. To numerically verify Line 10, we use bisection search to look for $\widehat{\theta}_i$, and abort the search when the search interval is small enough. We cannot prove that the IHT algorithm heuristic converges, but we observe that it converges to a solution of (16) in most of our numerical studies.

2) Stage 2 (buffer allocation, i.e., fix $\left(\boldsymbol{c}^d, \boldsymbol{c}^m, \boldsymbol{c}, \boldsymbol{\theta}\right)$*, solve for* $\left(\boldsymbol{n}^d, \boldsymbol{B}\right)$*):* This stage is similar to stage 2 in Sect. III-C.2 (by replacing $\boldsymbol{c}$ with $\boldsymbol{c}^d$), thus is omitted.

The AM-M algorithm iterates through two stages until $\left(\boldsymbol{n}^d, \boldsymbol{c}, \boldsymbol{B}\right)$ converge. The complete algorithm is given in Algorithm 4:

Algorithm 4 AM-M: Alternate Maximization for Mixed Data and Multimedia Traffic

repeat
    Determine $\left(\boldsymbol{n}^d, \boldsymbol{n}^m, \boldsymbol{c}^d, \boldsymbol{c}^m, \boldsymbol{c}, \boldsymbol{\theta}\right)$ under fixed $\boldsymbol{B}$: solve Problem (17) using IHT algorithm.
    Determine $\left(\boldsymbol{n}^d, \boldsymbol{B}\right)$ under fixed $\left(\boldsymbol{c}^d, \boldsymbol{c}^m, \boldsymbol{c}, \boldsymbol{\theta}\right)$: solve Problem (7) using bi-section search.
until Convergence

The AM-M algorithm can be modified to accommodate minimum bandwidth requirements for multimedia flows. This is done by adding the following constraints into problem (16): $c_i^m \geq c_{i,\min}^m, \forall i$, and changing the second constraint in (16) to $n_i^m M_i'(\theta_i) / M_i(\theta_i) = c_i^m - c_{i,\min}^m, \forall i$. In other words, we guarantee a positive $c_{i,\min}^m$ to user i in each scheduling interval.

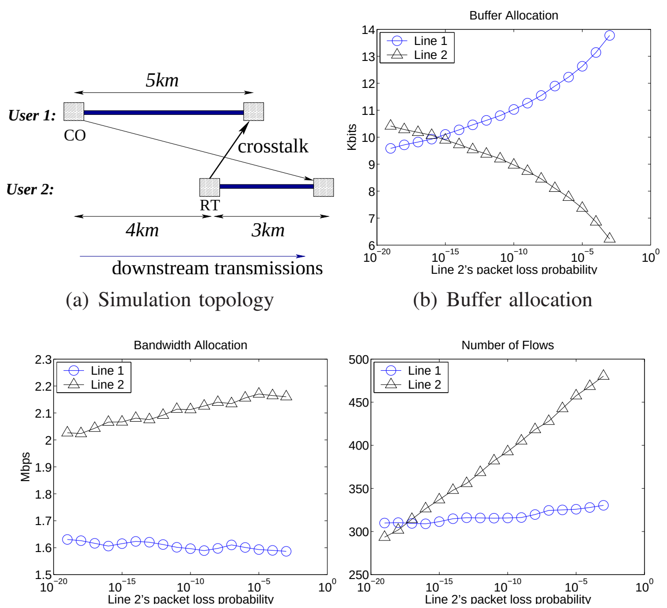
We note that the AM-D algorithm and AM-M algorithm apply to both downstream and upstream transmissions.

V. Numerical Examples

In this section, we demonstrate the performance of the proposed algorithms through a set of numerical examples. We first describe our simulation model for a DSL network. The scenario consists of two downstream transmitting ADSL modems as shown in Fig. 4(a), where the CO line (user 1) is 5 km long and the RT line (user 2) is 3 km long. The RT is deployed 4 km downstream from the CO. We use a realistic simulator with channel gains measured from actual DSL networks. ANSI noise model A [20] is used, which consists of 16 ISDN, 4 HDSL and 10 conventional (non-DSM capable) ADSL disturbers in the background.

Throughout this section, we assume that both lines' data traffic follow the same compound Poisson distribution with exponential file size (as discussed in Sect. III-A). The arrival rate λ equals 40 burst/sec and average file size $1/\mu$ equals 100 bits/burst. We consider the downstream transmission case, where a total of 20 Kbits buffer space is shared by two copper lines at the multiplexing link (not shown in the figure).

In Figs. 4(b) to 4(d), we plot the buffer allocations, bandwidth allocations and total admitted data flows for both users. Here the weights of two users are fixed at $(w_1, w_2) = (2, 1)$, and the packet loss probability of user 1 is fixed at $\epsilon_1^d = 10^{-10}$. We vary ϵ_2^d from 10^{-19} to 10^{-3}. Fig. 4(b) shows that with a higher value of ϵ_2^d, user 2 has a less stringent QoS constraint, thus user 2 requires less buffer space to prevent packet loss.

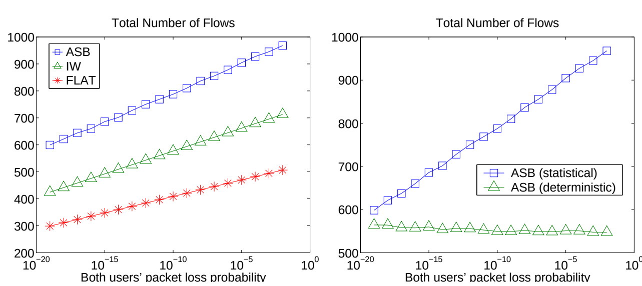


(a) Simulation topology (b) Buffer allocation

(c) Bandwidth allocation (d) Total admitted flows

Fig. 4. Bandwidth and buffer allocations as well as admitted flows for a two-user DSL network.

Also, since user 2's effective bandwidth decreases with ϵ_2^d, it is beneficial to allocate more bandwidth to user 2, since each unit of bandwidth can now support more flows from user 2 and thus contribute more to the overall objective in Problem (5). This is verified in Fig. 4(c). Fig. 4(d) shows that more flows from user 2 are admitted to the network. For user 1, since the buffer allocation increases (Fig. 4(b)) and bandwidth remains roughly unchanged (Fig. 4(c)) with ϵ_2^d, the total number of admitted flows also increases in Fig. 4(d).



(a) Comparison between different spectrum management algorithms. (b) Statistical multiplexing gain over deterministic services.

Fig. 5. Performance gain in the 2 user case.

Besides the ASB algorithm, other spectrum management algorithms can be used to solve the weighted rate maximization problem (4), including the Iterative Water-filling (IW) algorithm [12] (an earlier DSL spectrum management algorithm) and the FLAT algorithm (i.e., flat power allocation across all frequencies with no dynamic spectrum management, which is often used in practice today). Different DSL algorithms lead to different performances of the AM-D and AM-M algorithm. Figs. 5(a) and 5(b) are also obtained based on network topology in Fig. 4(a), and here we fix $(w_1, w_2) = (1, 1)$ and vary $\epsilon \left(= \epsilon_1^d = \epsilon_2^d\right)$ from 10^{-19} to 10^{-2}. Fig. 5(a) shows that the number of admitted flows by using ASB to solve Problem (4) is as much as 150% of that admitted by the IW-based algorithm, and 200% of that admitted by the FLAT based algorithm. Fig. 5(b) compares the performances between statistical and deterministic services. In the deterministic service, the flows are admitted based on their peak rates[4]. By allowing small probabilistic packet loss, the statistical service achieves a statistical multiplexing gain (measured in the ratio of total number of admitted flows) up to 180%.

To illustrate the statistical multiplexing effect of data and multimedia flows using the AM-M algorithm, we use a multimedia traffic model with average flow rate of 0.25Mbps for all users. For simplicity, we set the same packet loss probability for all flows. In addition, a minimum total bandwidth allocation for multimedia flows is set to 0.2Mbps. As shown in Fig 6, the multimedia flows for each line are guaranteed at the minimum rate, and the multimedia bandwidth for Line 2 tends to increase as the packet loss probability decreases without affecting the bandwidth allocation of the data flows.

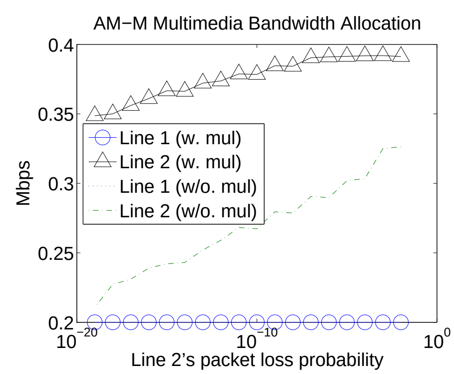


Fig. 6. Performance gain in statistical multiplexing for multimedia flows using the AM-M algorithm.

VI. Conclusions

In this paper, we consider the statistical multiplexing problem in DSL broadband access networks, where users have mixed data and multimedia traffic flows. We proposed a class of Alternate Maximization (AM) algorithms (AM-D and AM-M), which solve the problem by jointly allocating bandwidth and buffer resources to users. For data traffic, we simplify the problem using the concept of effective bandwidth, which captures the traffic characteristics, the QoS requirements, and users interactions at the physical layer. The corresponding problem can be solved using the ASB spectrum management algorithm as the core machinery. Numerical tests show that our algorithms exhibit good convergence, and admit as much as 200% of the number of data flows compared to a system configuration without spectrum management. The statistical multiplexing gain is up to 180% based on a simple two-line network topology. For multimedia traffic, we estimate

[4]We generate the compound Poisson traffic for 20000 seconds, and use the empirical peak rate as the basis for deterministic service.

the resource consumption based on a bufferless model and Chernoff bound, and proposed an Iterative Hypothesis Testing (IHT) algorithm to determine the amount of resource needed. This work is part of the FAST Copper Project that shows the promise of better broadband, ubiquitous access networks through research innovations in the dimensions of Frequency, Amplitude, Space, and Time.

Appendix: ASB Algorithm

The autonomous spectrum balancing (ASB) algorithm [15] is developed to solve Problem (4), which has an alternative representation as optimization over transmission power instead of rates. Consider all users transmitting over a total of L frequency tones. Denote user i's achievable rate on tone l as c_i^l, which is a function of the transmission power of all users on this tone (due to interference) $\boldsymbol{s}^l = \left(s_1^l, ..., s_N^l\right)$, the direct and crosstalk channel gains, and the background noise. The feasible power set for user i is defined as $\mathcal{S}_i = \left\{\boldsymbol{s}_i = \left(s_i^l, \forall l\right) \mid \sum_l s_i^l \leq S_i^{\max}, s_i^l \geq 0, \forall l\right\}$. Problem (4) can be represented as follows, with $\boldsymbol{s} = \left(s_i^l, \forall i, l\right)$:

$$\begin{aligned} \text{maximize} \quad & \sum_i \overline{w}_i \sum_l c_i^l\left(\boldsymbol{s}^l\right) \qquad (18) \\ \text{subject to} \quad & \boldsymbol{s}_i \in \mathcal{S}_i, \\ \text{variables} \quad & \boldsymbol{s} \geq \mathbf{0}. \end{aligned}$$

The optimal solution of Problem (18) determines a point on the Pareto boundary of the maximum rate region $\mathcal{C}$, and the complete boundary can be traced out with different weights $\overline{\boldsymbol{w}}$ [11]. Solving Problem (18) precisely is difficult since it is a nonconvex and tightly coupled problem. The ASB algorithm solves Problem (18) *approximately*, can be implemented autonomously and with low complexity (linear in N and L). The crux of the ASB algorithm is the "reference line" idea, which is a static *virtual* line that mimics the weakest user in the network and serves as a *static* penalty term for the actual users. It has *fixed* and *publicly* known parameters (i.e., transmission power, background noise, and crosstalk channel gains). Denote the reference line's achievable rate on tone l *from user i's perspective* as $c_i^{l,\text{ref}}\left(s_i^l\right)$, which is decreasing in user i's transmission power s_i^l due to the interference generated by user i.

The ASB algorithm is given in Algorithm 5, where κ_i is the Lagrangian dual variable for user i's power constraint, ε_i is a small stepsize, and $[x]^+ = \max\{x, 0\}$. In line 5, user i chooses the transmission power s_i^l to maximize a weighted sum of its own rate c_i^l and the reference line rate $c_i^{l,\text{ref}}$, subject to a penalty term proportional to κ_i.

We use the following result in [15] in this paper.

Lemma 2: Using the ASB algorithm, if user i's weight $\overline{w}_i$ increases while all other users' weights are fixed, the objective of Problem (18) always increases.

Algorithm 5 ASB: Autonomous Spectrum Balancing Algorithm

1: **repeat**
2:   **for all** each user $i = 1, ..., N$ **do**
3:     **repeat**
4:       **for all** each tone $l = 1, ..., L$ **do**
5:         $s_i^l = \arg\max_{\tilde{s}_i^l} \overline{w}_i c_i^l\left(\tilde{s}_i^l\right) + c_i^{l,\text{ref}}\left(\tilde{s}_i^l\right) - \kappa_i \tilde{s}_i^l.$
6:       **end for**
7:       $\kappa_i = \left[\kappa_i + \varepsilon_i\left(\sum_l s_i^l - S_i^{\max}\right)\right]^+.$
8:     **until** Convergence
9:   **end for**
10: **until** Convergence

References

[1] D. Wrege, E. Knightly, H. Zhang, and J. Liebeherr, "Deterministic delay bounds for VBR video in packet-switching networks: fundamental limits and practical trade-offs," *IEEE/ACM Trans. on Networking*, vol. 4, no. 3, pp. 352–362, 1996.

[2] R. Boorstyn, A. Burchard, J. Liebeherr, and C. Oottamakorn, "Statistical service assurances for traffic scheduling algorithms," *IEEE Journal on Selected Areas in Commun.*, vol. 18, no. 12, pp. 2651–2664, 2000.

[3] F. Presti, Z. Zhang, J. Kurose, and D. Towsley, "Source Time Scale and Optimal Buffer/Bandwidth Tradeoff for Heterogeneous Regulated Traffic in a Network Node," *IEEE/ACM Trans. on Networking*, vol. 7, no. 4, 1999.

[4] M. Reisslein, K. Ross, and S. Rajagopal, "A framework for guaranteeing statistical QoS," *IEEE/ACM Trans. on Networking*, vol. 10, no. 1, pp. 27–42, 2002.

[5] T. Bonald, A. Proutiere, J. Roberts, and F. R&D, "Statistical performance guarantees for streaming flows using expedited forwarding," *IEEE INFOCOM*, vol. 2, 2001.

[6] K. Kumaran and M. Mandjes, "Multiplexing regulated traffic streams: design and performance," *IEEE INFOCOM*, vol. 1, 2001.

[7] H. Chaskar, U. Madhow, N. Center, and M. Boston, "Statistical multiplexing and QoS provisioning for real-time traffic on wireless downlinks," *IEEE Journal on Selected Areas in Commun.*, vol. 19, no. 2, pp. 347–354, 2001.

[8] F. Yu and V. Krishnamurthy, "Effective bandwidth of multimedia traffic in packet wireless CDMA networks with LMMSE receivers: a cross-layer perspective," *IEEE Trans. on Wireless Commun.*, vol. 5, no. 3, pp. 525–530, 2006.

[9] J. Chong, B. Jung, and D. Sung, "Statistical Multiplexing-Based Hybrid FH-OFDMA System for OFDM-Based UWB Indoor Radio Access Networks," *IEEE Trans. on Microwave Theory and Techniques*, vol. 54, no. 4, pp. 1793–1801, 2006.

[10] V. Huang and W. Zhuang, "QoS-oriented packet scheduling for wireless multimedia CDMA Communications," *IEEE Trans. on Mobile Computing*, vol. 3, no. 1, pp. 73–85, 2004.

[11] R. Cendrillon, W. Yu, M. Moonen, J. Verlinden, and T. Bostoen, "Optimal multiuser spectrum balancing for digital subscriber lines," *IEEE Trans. on Commun.*, vol. 54, no. 5, pp. 922– 933, May 2006.

[12] W. Yu, G. Ginis, and J. Cioffi, "Distributed multiuser power control for digital subscriber lines," *IEEE Journal on Selected Areas in Commun.*, vol. 20, no. 5, pp. 1105–1115, June 2002.

[13] R. Cendrillon and M. Moonen, "Iterative spectrum balancing for digital subscriber lines," in *IEEE ICC*, 2005.

[14] R. Lui and W. Yu, "Low-complexity near-optimal spectrum balancing for digital subscriber lines," *IEEE ICC*, 2005.

[15] R. Cendrillon, J. Huang, M. Chiang, and M. Moonen, "Autonomous spectrum balancing for digital subscriber lines," to appear in *IEEE Transactions on Signal Processing*, 2007.

[16] G. Kesidis, J. Walrand, and C. Chang, "Effective bandwidths for multiclass Markov fluids and other ATM sources," *IEEE/ACM Trans. on Networking*, vol. 1, no. 4, pp. 424–428, 1993.

[17] F. Kelly, "Notes on effective bandwidths," *Stochastic Networks: Theory and Applications*, vol. 4, pp. 141–168, 1996.

[18] S. Boyd and L. Vanderberghe, *Convex Optimization*. Cambridge University Press, 2004.

[19] Z. Zhang, J. Kurose, J. Salehi, and D. Towsley, "Smoothing, statistical multiplexing, and call admission control for stored video," *IEEE Journal on Selected Areas in Commun.*, vol. 15, no. 6, pp. 1148–1166, 1997.

[20] V. Oksman and J. M. Cioffi, "Noise models for vdsl performance verification," ANSI, ANSI-77E7.4/99.438R2, Dec. 1999.