FAST Copper For Broadband Access

Mung Chiang[1], Jianwei Huang[1], Raphael Cendrillon[2], Chee Wei Tan[1], Dahai Xu[1]
[1] Electrical Engineering Department, Princeton University, USA [2] Marvell Semiconductors

1. INTRODUCTION TO THE FAST COPPER PROJECT

FAST Copper is a multi-year, U.S. NSF funded project that started in 2004, and is jointly pursued by the research groups of Mung Chiang at Princeton University, John Cioffi at Stanford University, and Alexader Fraser at Fraser Research Lab, and in collaboration with several industrial partners including AT&T. The goal of the FAST Copper Project is to provide ubiquitous, 100 Mbps, fiber/DSL broadband access to everyone in the U.S. with a phone line. This goal will be achieved through two threads of research: dynamic and joint optimization of resources in Frequency, Amplitude, Space, and Time (thus the name 'FAST') to overcome the attenuation and crosstalk bottlenecks, and the integration of communication, networking, computation, modeling, and distributed information management and control for the multi-user twisted pair network.

Access networks are often the rate-reach-QoS bottleneck of end-to-end connections in wide area networks. Realizing the vision of truly broadband and ubiquitous access to almost everyone in the U.S. is a formidable task, with many significant technical and socio-economic challenges. Although the fiber-to-the-home solutions promise to provide broadband delivery, the labor costs associated with fiber installation need to be divided over the number of customers served by the fiber. Such cost becomes increasingly expensive as the number of customers served decreases, which happens when fiber gets closer and closer to the customer, especially in suburban areas. That last segment labor cost of fiber is the dominant limitation in broadband access.

We propose to leverage the installed copper plant, which is by far the most ubiquitous access network in the U.S. The overall architecture is a hybrid fiber/DSL deployment. To achieve data rates significantly higher than the current levels on low-twist unshielded telephone wires demands thinking about transmission on copper wires in a new way. This project combines innovative optimization and signal processing techniques with novel system architecture and protocols, as well as an integrated plane of real-time control, computation, data collection, and auto-configuration, to enable an access infrastructure that is *both* broadband and ubiquitous.

Traditionally, DSL broadband access networks have been analyzed by viewing each twisted pair as a separate communication channel, independent of other twisted pairs in the same binder cable. We believe that the key to realizing the vision of ubiquitous, readily deployable, and truly broadband access networks is to dynamically optimize the resources in the dimensions of Frequency, Amplitude, Space, and Time, in the multiple-input-multiple-output communication environment of DSL across *multiple layers* in the protocol stack.

The key idea behind the FAST framework is that, instead of holding the traditional view that each twisted pair is an independent channel, we model a bundled cable of twisted pairs as one aggregate *multi-user* communication system. Multiple users *compete* against and *cooperate* with each other in this system. The basic premise of FAST is to explicitly take into account the *crosstalk* effects (both near-end and far-end) that currently form the data rate bottleneck, and to exploit potential *cooperation* in sharing limited resources in all four dimensions of F, A, S, and T:

- In the physical layer, new techniques can be developed based on improving spectral utilization, mitigating multi-user interference, and exploiting multi-user cooperation. Through dynamic adaptation and utilization of frequency spectrum, such as power control, bit loading, or vectored transmission, Dynamic Spectrum Management (DSM)[24] allows maximum flexibility in allocating rates among competing flows, achieves much higher total data rates, and extends the reach of broadband access.
- FAST Copper will also leverage the potential for time division multiplexing based on the application layer burstiness of data traffic from and to the end hosts. In most communication-theoretic investigations, it is assumed that there is always an infinite backlog of bits that need to be transmitted per user, thus taking out the latency considerations and the temporal dimension. By jointly considering the application layers, burstiness of the required bandwidth provides another degree of flexibility of statistical multiplexing along

the temporal axis. FAST Copper will investigate how to complement FDM-based DSM in the physical layer and TDM-based scheduling techniques in the MAC layer.

- 'Space' is yet another important dimension where resources must be optimized. When building robust and efficient broadband access networks, two issues are particularly important: how can a hybrid fiber/twisted pair architecture be designed to utilize the best of fiber-based and copper-based communication potentials, and how can a logical topology be designed to offer fast-recovery after natural failures or malicious attacks?
- We propose to install active 'amplitude control' mechanisms to shape the flow intensities at the edge to provide different QoS classes through dynamic bandwidth allocation. At the same time, a network management system constantly probes, measures and monitors the cable and its environments, receives data rate requests from user terminals, and periodically shapes the rate each user is allowed to transmit and receive per time frame.

Offering 100 Mbps data rate over twisted-pair presents tremendous technical challenges. We must revolutionize both the digital signal processing algorithms in the physical layer and the architecture/protocol design methodologies in the 'upper' layers. We also need to carefully investigate the coupling effects across multiple layers so that end-user experience over broadband access networks is enhanced, which can be conducted using the recently developed framework of "Layering As Optimization Decomposition"[9] for layered network architecture.

In summary, by modeling the whole binder of copper wires as one MIMO channel, with resources ranging from the physical layer to the application layer, we can dynamically optimize resource allocations over Frequency, Amplitude, Time, and Space, in a stable, robust, and complementary way. Collectively, these four degrees of freedom offer many exciting opportunities to make practical impacts. At the same time, progress in the project come from solving important problems in the fundamental research disciplines of information theory, signal processing, nonlinear optimization, distributed control, network protocol design, and graph theory.

This paper provides a brief overview of some of the latest developments this year at Princeton University for this actively ongoing FAST Copper project. This overview is presented along the dimensions of Frequency, Time, Amplitude, and Space, with more mature results for the Frequency axis of the project.

2. FREQUENCY

2.1. Introduction

There are two major obstacles for performance improvement in modern DSL systems: attenuation and crosstalk (i.e., interference generated between different lines in the same binder). While attenuation will be mitigated through a hybrid fiber/DSL architecture where DSL is responsible for the 4-6 kft, crosstalk can be mitigated through mechanisms that encourages multi-user cooperation through spectrum management in frequency and scheduling in time. This section focuses on various Dynamic Spectrum Management (DSM) methods.

The crosstalk is typically 10-20 dB larger than the background noise, and direct crosstalk cancelation (e.g.,[3,11]) may not be feasible in many cases due to the complexity issues or as a result of unbundling. In the case of perfect synchronization between the different Discrete MultiTone (DMT) transmission blocks, the crosstalk experienced by a line on a certain tone is due to the transmissions of other lines on the same tone. In practice, however, perfect DMT synchronization could be difficult to achieve due to differences in channel propagation delays. In that case, orthogonality among tones are destroyed and inter-carrier-interference (ICI) leads to more serious crosstalks. In both the synchronous and asynchronous cases, DSM can significantly improve data rates over the current practice of static spectrum management that mandates spectrum mask or flat power backoff across all frequencies (i.e., tones). In particular, we will show in this section a suite of algorithms for power allocation (or, equivalently, bit loading) for DSL networks, called Autonomous Spectrum Balancing (ASB). All versions of ASB are autonomous (distributed algorithm across the users without explicit information exchange) with low complexity, while provably convergent and comes close to the globally optimal rate region in practice, thus overcoming bottlenecks in the state-of-the-art DSM algorithms as discussed below.

Most of the recently proposed DSM algorithms focus on the synchronous transmission case, including the *Iterative Water-filling* (IW) algorithm,[28] the *Optimal Spectrum Balancing* (OSB) algorithm,[5] and the *Iterative Spectrum Balancing* (ISB) algorithm.[4,18] In the IW algorithm, each line maximizes its own data rate by waterfilling over the noise and interference from other lines. The IW algorithm is a completely autonomous algorithm with a linear complexity in the number of users. Unfortunately, although IW can achieve near optimal performance in weak interference channels, it is highly-suboptimal in near-far scenarios, such as mixed central office (CO)/remote terminal (RT) deployments of ADSL and upstream VDSL, because of the greedy nature of the algorithm. The OSB algorithm addresses this problem through maximization of a weighted rate-sum that explicitly takes into account the damage done to the other lines within the network when optimizing each line's spectra. Unfortunately OSB has an exponential complexity in the number of users, making it extremely complex when the DSL system contains many lines. Furthermore, the OSB algorithm is not distributed, instead relying on a centralized network management center (NMC) to optimize the PSDs for all modems. This NMC requires knowledge of the crosstalk channels between all lines, something that is often not available in practice. This is because, for example, the regulatory requirements on "unbundling" service, i.e., incumbent service providers must rent certain lines to their competitors. This makes it very costly to perform a centralized optimization. Also, many lines in the same binder terminate on different quad cards in the DSL Access Multiplexer because customers in the same neighborhood sign up at different times, which makes it impossible to have central coordination even if one can tolerate the cost issues. In recent work, ISB was proposed, which implements the weighted-rate sum optimization of OSB in an iterative fashion over the users. This leads to a quadratic complexity in the number of users, however ISB still requires centralized operation. All these algorithms utilize the dual-based decomposition technique by relaxing modem's individual power constraints and making the spectrum balancing problem separable across tones. As a result, they are not directly applicable in the asynchronous transmission case, since dual-based relaxation here will not make the problem separable due to the additional coupling across tones caused by ICI.

For the asynchronous transmission case, the author in[7] proposed two centralized greedy power allocation algorithms, *bit-subtracting* and *bit-adding* algorithms. Both algorithms start from the power spectrum density (PSD) obtained with the ISB algorithm in the synchronous case and search for local optimal solutions in the neighborhood by taking ICI into account. Due to the centralized nature of these algorithms, they are computational expensive in the case of large numbers of users and tones.

The suite of ASB algorithms recently developed[13] has the following advantages compared with the previous algorithms. First of all, ASB is autonomous: it can be applied in a distributed fashion across users with no explicitly information exchange. Furthermore, the algorithm has low complexity in both the number of users and tones, and is provably convergent under reasonable conditions on the channel gains that are often satisfied in DSL. In the synchronous case, ASB algorithms (including ASB-1 and ASB-2) achieve similar complexity as IW, but achieves much better performance than IW and close to ISB and OSB. In the asynchronous case, ASB algorithms, reduce the complexity from those in,[7] and achieve significant better performance than the ASB algorithm that do not consider the ICI.

The basic idea behind ASB is to leverage the fact that DSL interference channel gains are very slowly time-varying, which enables an effective use of the concept of "reference line" that represents a typical victim within a DSL system. When adapting its PSD, each line attempt to achieve its own target rate whilst minimizing the damage it does to the reference line, thereby achieving a reasonable balance between selfish and socially responsible operation. We prove the convergence of ASB under an arbitrary number of users, for both sequential and parallel updates. Since IW can be recovered as a special case of ASB in the synchronous case, our work extends previous work on IW.[10,28] In this paper we will briefly discuss the ASB algorithm for the synchronous transmission, and Table 1 compares various aspects of different DSM algorithms in the synchronous case, where ASB attains the best tradeoff among distributiveness, complexity, and performance. Here we use K to denote the number of tones and N to denote the number of users. For details on the analysis, proofs as well as ASB algorithm for the asynchronous transmission, see.[6,13]

2.2. System Model

Consider a DSL network with a set $\mathcal{N} = \{1, ..., N\}$ users (i.e., lines, modems) and $\mathcal{K} = \{1, ..., K\}$ tones (i.e., frequency carriers). Assuming the standard *synchronous* discrete multi-tone (DMT) modulation is applied,

Table 1. Comparison of DSM algorithms in the synchronous transmission case

Algorithm	Operation	Complexity	Performance	Reference
IW	Autonomous	$O(KN)$	Sub-optimal	[28]
OSB	Centralized	$O\left(Ke^{N}\right)$	Optimal	[5]
ISB	Centralized	$O\left(KN^{2}\right)$	Near optimal	[4,18]
ASB-1/2	Autonomous	$O(KN)$	Near optimal	[13]

transmission can be modeled independently on each tone k as $\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{z}_k$. The vector $\boldsymbol{x}_k \triangleq \{x_k^n, n \in \mathcal{N}\}$ contains transmitted signals on tone k, where x_k^n is the signal transmitted by user n at tone k. Vectors $\boldsymbol{y}_k$ and $\boldsymbol{z}_k$ have similar structures: $\boldsymbol{y}_k$ is the vector of received signals on tone k, and $\boldsymbol{z}_k$ is the vector of additive noise on tone k and contains thermal noise, alien crosstalk and radio frequency interference. We denote the channel gain from transmitter m to receiver n on tone k as $h_k^{n,m}$. We denote the transmit power spectrum density (PSD) $s_k^n \triangleq \mathcal{E}\left\{|x_k^n|^2\right\}$, where $\mathcal{E}\{\cdot\}$ denotes expected value. The vector containing the PSD of user n on all tones as $\boldsymbol{s}^n \triangleq \{s_k^n, k \in \mathcal{K}\}$.

Assume that each user treats interference from other modems as noise. When the number of interfering users is large, the interference can be well approximated by a Gaussian distribution. Under this assumption the achievable bit rate of user n on tone k is

$$b_k^n \triangleq \log\left(1 + \frac{1}{\Gamma}\frac{s_k^n}{\sum_{m \neq n} \alpha_k^{n,m} s_k^m + \sigma_k^n}\right), \tag{1}$$

where $\alpha_k^{n,m} = |h_k^{n,m}|^2 / |h_k^{n,n}|^2$ is the normalized crosstalk channel gain (with $\alpha_k^{n,n} = 0, \forall k, n$), and σ_k^n is the noise power density normalized by the direct channel gain $|h_k^{n,n}|^2$. Here Γ denotes the SINR-gap to capacity, which is a function of the desired BER, coding gain and noise margin.[23] For notational simplicity, we absorb Γ into the definition of $\alpha_k^{n,m}$ and σ_k^n (i.e., let $\Gamma = 1$). The bandwidth of each tone is normalized to 1. Each user n is typically subject to a total power constraint P^n, due to the limitations on each modem's analog frontend: $\sum_{k \in \mathcal{K}} s_k^n \leq P^n$. The data rate on line n is thus $R^n = \sum_{k \in \mathcal{K}} b_k^n$.

The spectrum management problem is defined as follows

$$\max_{\{\boldsymbol{s}^n, n \in \mathcal{N}\}} \sum_n w^n R^n \text{ s.t. } \sum_{k \in \mathcal{K}} s_k^n \leq P^n, \forall n. \tag{2}$$

where w^n is a nonnegative weight coefficient of user n. Due to interference between lines, Problem (2) is nonconvex. Furthermore, it is highly coupled across lines (due to crosstalk) and tones (due to total power constraint), making it a very difficult optimization to solve. In particular, any algorithm that globally solves (2) must have knowledge of all crosstalk channels and background noise spectra, forcing it to operate in a centralized fashion. In order to overcome this difficulty, we observe that for optimal solutions of (2) each user adopts a PSD that achieves a fair compromise between maximizing their own data-rate and minimizing the damage they do to other lines within the network.

Based on this insight, we introduce the concept of a "reference line", a virtual line that represents a "typical" victim within the DSL system. Since network operators are typically concerned with maximizing the rate achieved by the worst line within their network, the reference line typically corresponds to the longest line in the network (e.g. the CO distributed line in a mixed CO/RT scenario, such as that in Section 2.5), which has the weakest direct channel and receives relatively stronger crosstalk from other users. Then instead of solving (2) , each user tries to maximize the achievable rate on the reference line, subject to its own rate and total power constraints.

Since the main purpose of introducing the reference line is to characterize the "damage" that each user does to the network, we will make the achievable rate of the reference line user dependent. In other words, *from user* *n's point of view*, the reference line's rate is $R^{n,ref} \triangleq \sum_{k\in\mathcal{K}} \tilde{b}_k^n$, where the achievable bit rate on tone k is defined as

$$\tilde{b}_k^n \triangleq \log\left(1 + \frac{\tilde{s}_k}{\tilde{\alpha}_k^n s_k^n + \tilde{\sigma}_k}\right), \tag{3}$$

The coefficients $\{\tilde{s}_k, \tilde{\sigma}_k, \tilde{\alpha}_k^n, \forall k, n\}$ are parameters of the reference line and can be obtained from long-term field measurements*. Since the crosstalk channel can be regarded as time-invariant in the DSL wireline network, the parameters of the reference lines are known to users a priori. Intuitively, the reference line serves a penalty term in each user's optimization problem to avoid purely selfish behavior, and eliminates the need of explicit message passing amongst users.

Thus instead of solving Problem (2) which requires global information, we let each user n solve the following problem in ASB algorithm (treating the crosstalks from other users as fixed Gaussian noise),

$$\max_{\boldsymbol{s}^n} w^n R^n + R^{n,ref} \quad \text{s.t.} \sum_{k\in\mathcal{K}} s_k^n \le P^n. \tag{W-SUM}$$

In other words, we let each user solve a problem locally, treating the reference line as a "static pricing" term. Users then iterate until PSD converges.

2.3. ASB Algorithms

The ASB algorithm involves two levels. In the first level, we use a dual-based approach to decompose Problem (W-SUM) into one subproblem on each tone. This involves relaxing the power constraint using a dual variable. Although Problem (W-SUM) is nonconvex, we know[5] that the corresponding duality gap of Problem (W-SUM) is asymptotically zero (when the number of tones goes to infinity), thus solving the dual problem can lead to optimal primal solution.

The main difference among various ASB algorithms lies in the second level, i.e., how to solve the subproblem on each tone. Especially, ASB-1 algorithm solves a nonconvex subproblem on each tone by exhaustive search, and ASB-2 solves a convex subproblem based on the high SINR relaxation of the reference line. Each user will adjust the dual variable to make the power constraint tight, based on the solution on each tone. Then users take turns to perform this optimization until the PSDs converge.

2.3.1. ASB Base Algorithm (ASB-1)

By incorporating the total power constraint into the objective function, we have the following Lagrangian of Problem (W-SUM),

$$L^n \triangleq w^n R^n + R^{n,ref} - \lambda^n \sum_{k\in\mathcal{K}} s_k^n$$

Here λ^n represents the dual variable of user n and needs be chosen such that $\sum_k s_k^n = P^n$ or $\lambda^n = 0$. Then Problem (W-SUM) can be solved by the following unconstrained optimization problem,

$$\max_{\boldsymbol{s}^n} L^n\left(w^n, \lambda^n, \boldsymbol{s}^n, \boldsymbol{s}^{-n}\right), \tag{4}$$

where $\boldsymbol{s}^{-n} = \{\boldsymbol{s}_k^m, \forall m \ne n\}$ denotes the PSD of all users except user n. Further define

$$L_k^n = w^n b_k^n + \tilde{b}_k^n - \lambda^n s_k^n, \tag{5}$$

then it is clear that L^n can be decomposed into a sum across tones of L_k^n, $L^n = \sum_k L_k^n$. As a result, Problem (4) can be decomposed into K subproblems, one for each tone k. The optimal PSD that maximizes L_k^n is

$$s_k^{n,1} = \arg \max_{s_k^n \in [0, P^n]} L_k^n\left(w^n, \lambda^n, s_k^n, s_k^{-n}\right), \tag{6}$$

*In fact, the reference line concept is already used in existing VDSL standards such as T1.424-2004. Good choices for reference lines have been defined based on extensive studies. However, it has not been used for PSD optimization as proposed in the ASB algorithm.

where $s_k^{-n} = \{s_k^m, \forall m \neq n\}$. Since L_k^n is not a convex function in s_k^n, the optimal value $s_k^{n,S1}$ can be found as follows. First solve the first order condition, $\partial L_k^n / \partial s_k^n = 0$, which leads to a cubic equation which has three roots. Then compare the value of L_k^n at each of these three roots, as well as checking the boundary solutions $s_k^n = 0$ and $s_k^n = P^n$, we can find out the value of $s_k^{n,1}$.

User n then updates λ^n to enforce the total power constraint. Users then iterate until all the PSD converge.

2.3.2. ASB with High SINR Approximation (ASB-2)

We now introduce a variation of the ASB algorithm (ASB-2) that enjoys even lower computational complexity and has provable convergence. Instead of solving cubic equation on every tone as in ASB-1, we assume that the reference line operates in the high SINR regime whenever it is active, that is iff $\tilde{s}_k > 0$, then $\tilde{s}_k \gg \tilde{\sigma}_k \gg \alpha_k^{n,m} s_k^n$ for any feasible s_k^n, $n \in \mathcal{N}$ and $k \in \mathcal{K}$. This assumption is motivated by our observations of optimal solutions for DSL interference channels. Intuitively, we assume that the received signal power on the reference line is much larger than the reference noise, which is in turn much larger than the interference from user n. Thus on any tone k where the reference line is active (i.e., $\tilde{s}_k > 0$), the corresponding achievable bit rate in (3) is approximated as

$$\tilde{b}_k^n = \log\left(1 + \frac{\tilde{s}_k}{\tilde{\alpha}_k^n s_k^n + \tilde{\sigma}_k}\right) \approx \log\left(\frac{\tilde{s}_k}{\tilde{\sigma}_k}\right) - \frac{\tilde{\alpha}_k^n s_k^n}{\tilde{\sigma}_k}. \tag{7}$$

By plugging (7) into (5) and solving similarly as in (6), we obtain the optimal PSD values as:

$$s_k^{n,2}\left(w^n, \lambda^n, s_k^{-n}\right) = \left[\frac{w^n}{\lambda^n + \tilde{\alpha}_k^n \mathbf{1}_{\{\tilde{s}_k > 0\}} / \tilde{\sigma}_k} - \sum_{m \neq n} \alpha_k^{n,m} s_k^m - \sigma_k^n\right]^+, \tag{8}$$

where $\mathbf{1}_{\{\bullet\}}$ as the indication function. This is essentially a water-filling type solution, with different water-filling levels for different tones. For this reason we term this algorithm *frequency selective waterfilling.*

2.4. Convergence Analysis

We first show the convergence of ASB-1 under fixed $\boldsymbol{\lambda}$.

THEOREM 2.1. *Consider a two-user network with fixed $\boldsymbol{\lambda}$. There exists at least one fixed point of ASB-1, and the algorithm converges if users start from initial PSD values $\left(s_k^1, s_k^2\right) = \left(0, P^2\right)$ or $\left(s_k^1, s_k^2\right) = \left(P^1, 0\right)$ on all tones.*

The proof of Theorem 2.1 uses supermodular game theory[25] and strategy transformation similar to[12] and is omitted here. The convergence in Theorem 2.1 does not require any condition on the crosstalk channels. However, it is only for the case of fixed $\boldsymbol{\lambda}$.

We then consider the convergence of ASB-2 algorithm. We consider both sequential and parallel updates. Denote $s_k^{n,t}$ as the PSD of user n on tone k after iteration t, where $\sum_k s_k^{n,t} = P^n$ is satisfied at the end of any iteration t for any user n. In the sequential updates, only one user will change its PSD at any time. One iteration is defined as one round of updates of all users. In the more realistic but harder-to-analyze parallel updates, time is divided into slots, and the users update their PSDs simultaneously in each time slot, where the λ^n is adjusted such that the power constraint is tight.

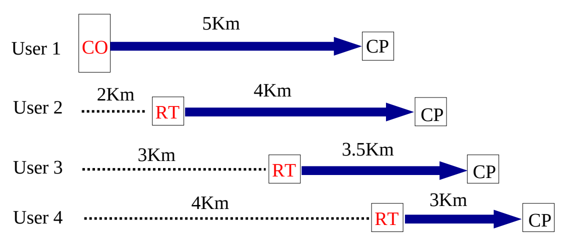
THEOREM 2.2. *Assume $\max_{m \neq n, k} \alpha_k^{n,m} < \frac{1}{N-1}$, then the ASB-2 algorithm globally and geometrically converges to the unique fixed point in an N-user system, with either sequential or parallel updates.*

Theorem 2.2 recovers the convergence of iterative water-filling in an N-user case with sequential updates (proved in[10]) as a special case. Moreover, the convergence proof for the parallel updates turns out to be simpler than that for sequential updates.

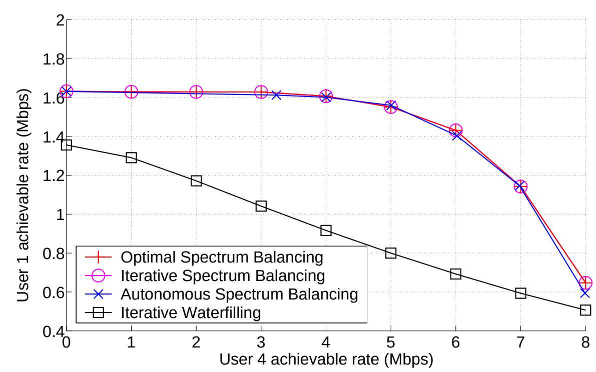
2.5. Simulation Results

Here we summarize a typical numerical example comparing the performance of the ASB-1 algorithms with IW, OSB and ISB in the synchronous transmission case. A four-user mixed CO/RT scenario has been selected to make a comparison with the highly complex OSB algorithm possible. As depicted in Fig. 1(a), user 1 is CO distributed, whilst the other three users are RT distributed. Due to the different distances among the corresponding transmitters and receivers, the RT lines generate strong interferences into the CO line, whilst experiencing very little crosstalk from the CO line. The target rates of users 2 and 3 have both been set to 2 Mbps (by adjusting the corresponding weights w^n). For a variety of different target rates of user 4, user 1 (the CO line) attempts to maximize its own data-rate either by transmitting at full power in IW, or by setting its corresponding weight w_{co} to unity in OSB, ISB and ASB-1. This produces the rate regions shown in Fig. 1(b), which shows that ASB-1 achieves near optimal performance similar as OSB and ISB, and significant gains over IW. For example, with a target rate of 1 Mbps on user 1, the rate on user 4 reaches 7.3 Mbps under ASB-1 algorithm, which is a 121% increase compared with the 3.3 Mbps achieved by IW.

We have also performed extensive simulations with different CO and RT positions, line lengths and reference line parameters. We found that the performance of ASB is very insensitive to definition of reference line: with a single choice of the reference line we observe good performance in a broad range of scenarios, and consistently significant gains over IW.



(a) A four-user mixed CO/RT deployment topology.



(b) Rate regions obtained by various DSM algorithms (ASB represents ASB-1).

Figure 1. Performance comparison of various dynamic spectrum management (DSM) algorithms.

3. TIME AND AMPLITUDE

3.1. Introduction

In the previous section, we have discussed spectrum management in DSL networks using ASB algorithm, with an objective of maximizing users' total weighted rates. We have made two implicit assumptions there: (1) all users always have packets to send (infinite backlog case), and (2) users' weights are fixed. In practice, however, traffic flows are stochastic in nature, which means that not all users transmit packets all time. In this case, the weights in the ASB algorithm need to be dynamically adjusted to reflect the relative priority and level of backlog of the users. A higher weight henceforth priority is associated with a user with a larger backlog. Furthermore, applications have different QoS requirements (e.g., loss rate or delay requirements), which affect how the weights are determined. Hence, traffic flow characteristics need to be jointly considered with dynamic spectrum management.

A worthwhile approach is to characterize the means to support high-speed transmission of different application traffic flows over the DSL network. High-speed burst transmission over the copper plant requires fast synchronization and equalization. Upstream scheduling for burst transmissions can use layer 2 protocols as a basis for supervision and control of upstream flows. Flow control at layer 2 provides a viable response to overload in a statistically mmltiplexed network. Access can be delayed or denied in order to protect the network and to allow the physical plant to stay within its design envelope. Downstream flows from the edge node to the end users must also be regulated so that the copper segment of the transmission path is not overwhelmed. Coordination of flows from different users has to take into account the length of the different loops as the length of the loop affects the tones utilized which in turn affects the achievable data rate of the end user.

Here we will consider the problem of joint statistical multiplexing, scheduling and spectrum management in DSL networks. Statistical multiplexing determines how much traffic a network can support over a relative large time scale subject to various QoS service requirements. Scheduling determines how to allocate resource (e.g., time slots, transmission rates) to different users over a relatively shorter time interval. Dynamic spectrum management determines how to support the scheduling decisions at every time instant. We want to design these three function modules jointly such that the communication resources are efficiently utilized, while still keeping intact the "layering" structure of the network protocol stack. In this paper, we develop a framework with the following main features:

- Maximize the application utilities of end-users with network stability and QoS performance guarantees.
- Exploit aggressive statistical multiplexing at the link level among different users.

We adopt a cross layer design approach where the achievable rate region of the DSL physical layer provides the capacity that supports the different traffic flow requirements of upper layers. A particular user application may give rise to a different minimum rate requirement and other objective function, i.e., network utility to be maximized. Furthermore, utility of allocated resources to end users and elasticity of application traffic can both be modeled through general utility functions. Utility functions also provide a measure of resource allocation efficiency and lead to allocations satisfying fairness definitions.[9] Furthermore, benefits of innovations in the physical layer through better coding, modulation or power allocation schemes can be characterized by the enhancement to applications rather than just the drop in bit error rates, which users do not directly observe.

We develop a practical joint spectrum management and scheduling algorithm as a throughput optimal control strategy for scheduling flows that meet peak power constraints at each modem while maximizing network utilities. The distributed algorithm also minimizes the average power expenditure in the whole access network in the long run. The implication is that the allocated power is used judiciously to maximize end user application utilities rather than being expended to overcome uncontrolled interference at the physical layer. Traditionally, the different DSL algorithms assumes a flow model where users transmit continuously at fixed data rate. However, in practice, traffic flows are bursty in nature due to the stochastic flow arrival distribution. By combining many users whose different application data rate is uncorrelated in time, substantial economies can be realized with the scheduling of aggregated bursty and constant rate flows. A novel idea in our joint spectrum management and scheduling approach is to map the traffic flow requirement at each scheduling interval as an optimization constraint in the utility maximization which is solved by the ASB algorithm iteratively. This is shown to be asymptotically optimal, i.e., maximize the total network utilities in the long run.

We install active 'amplitude control' mechanisms at the MAC layer to shape the flow intensities at the edge to provide different QoS classes through dynamic bandwidth allocation. Multiple flows may share a common time frame and transmit simultaneously but at different levels of allowed rates. This mechanism can be used to strike a balance between avoiding or mitigating crosstalk and meeting latency and fairness constraint.

The network model we now consider includes N DSL downstream transmissions that go through a same multiplexing link with total buffer size B. The QoS requirements of the applications depend on the bandwidth and buffer allocations. The bandwidth on each of the DSL link depends on DSM (e.g., ASB algorithm).

The approach we take is a network utility maximization framework[1] where users in the DSL network have rate adaptive traffic to send. Here "rate-adaptive" traffic includes both traditional data traffic (e.g., ftp downloading) and multimedia traffic with adaptive coding schemes, e.g., video streaming with smart summarization[14,17][†].

[†]We do not consider the inelastic traffic such as fixed quality video streaming, which is taken care of by admission control and subtract the required bandwidth from the available network capacities.

Each user n is associated with a utility function $U_n(T_n)$, which is continuous, increasing and strictly concave in the steady-state long-term empirical throughput T_n. Our goal is to determine how the communication resources, i.e., buffer, bandwidth, should be allocated such that the total utility $\sum_n U_n(T_n)$ is maximized. It is shown that a gradient-based scheduling approach leads to the unique optimal solution of maximizing network utilities in the long run.[1] This observation serves as our starting point. In other word, we want to develop throughput optimal dynamic algorithms at each statistical multiplexing and scheduling interval, where the weights in the ASB algorithm are determined dynamically by the gradients of end-users' utility functions.

3.2. Statistical Multiplexing

First, we consider the problem of statistical multiplexing at the link layer for downstream transmissions, which determines how much traffic to be admitted into the network such that resource is fully utilized subject to the QoS requirements. Previous work in statistical multiplexing have focused on the study of a single network node[21] or several cascade network nodes[22] in the wireline network. Some related work in the wireless network have proposed to using "link shaping" to transform the error-prone wireless channel into a nearly lossless link pipe.[8] In all these cases, the capacity of a link is assumed to be fixed regardless of the way in which it is shared among the users. This is however not the case in DSL networks where the link capacities are closely coupled among users due to crosstalk interferences. The interference-limited capacity region is nonconvex henceforth hard to determine, but nevertheless is typically convex in the asymptotic sense.[5] Statistical multiplexing at the link layer should explicitly take this into consideration.

3.2.1. Multiplexing Delay Insensitive Data Traffic

For delay insensitive data traffic of user n, the QoS requirement is defined as that the packet loss probability due to buffer overflow should be less than a given tolerance threshold ϵ_n. When ϵ_n is very small and the available buffer size B is very large, the bandwidth requirements of users' data traffic can be estimated accurately using the concept of *effective bandwidth*[16]. We can think of a stochastic data traffic of user n with average rate a_n and peak rate r_n as a constant rate traffic with effective bandwidth $\nu_n(\delta)$, where δ is a parameter that increases with ϵ_n and decreases with B. In general, we have $a_n \leq \nu_n(\delta) \leq r_n$, with $\nu_n(\delta) = a_n$ when δ approaches 0, and $\nu_n(\delta) = r_n$ when δ approaches ∞. This means that the effective bandwidth becomes closer to the average rate when the QoS requirement is not stringent and buffer size is large, and it will approach the peak rate when the QoS requirement becomes more stringent and the buffer size decreases. The specific function form of ν_i depends on the stochastic nature of user n's traffic (for more details on effective bandwidth, see for example the survey in[16]).

We want to determine how many flows of each user can be admitted by the network, $\boldsymbol{g} = \{g_n, n \in \mathcal{N}\}^{\ddagger}$. In other words, the average rate achieved by user n would be $g_n a_n$. Also each user n has a weight coefficient w_n, which equals the marginal utility at the scheduling time instants, $U_n'(T_n)$.

In the case where all users have the same QoS requirements, i.e., $\epsilon_n = \epsilon$ for all n, we let all users share the total buffer with size B. In the more general case where users have different ϵ_i, we may need to allocate a separate queue for each user n with different buffer size B_n with $\sum_n B_n = B$. In that case, the effective bandwidth of user n is $\nu_i(-\log \epsilon_i / B_i)$, where the specific function form of ν_i depends on the traffic model of user n (for more details, see[16]). Denote the capacity of DSL link n (user n) as c_n, then the total effective bandwidth of user n's traffic should satisfy $g_n \nu_n(-\log \epsilon_n / B_n) \leq c_n$. The value of vector $\boldsymbol{c} = \{c_n, n \in \mathcal{N}\}$ needs to be chosen from the feasible rate region, $\mathcal{C}$, which is determined by the crosstalk channel gains at the particular scheduling time instant and the solutions of the specific spectrum management algorithm. Mathematically, we have the following

‡We use bold symbols to denote vectors.

problem formulation:

$$\begin{aligned}
\max_{(\boldsymbol{g},\boldsymbol{c},\boldsymbol{B})\geq 0} \quad & \sum_n w_n a_n g_n \\
s.t. \quad & g_n \nu_n \left(-\log \epsilon_n / B_n\right) \leq c_n, \forall n \\
& \sum_n B_n = B, \\
& \boldsymbol{c} \in \mathcal{C},
\end{aligned} \tag{9}$$

where vector $\boldsymbol{B} = \{B_n, n \in \mathcal{N}\}$. Finding the global optimal solution of Problem (9) efficiently can be difficult since Problem (9) is typically non-convex. However, a feasible solution can nevertheless be computed with a two-stage iterative algorithm where we first fix $\boldsymbol{B}$ and solve $(\boldsymbol{g}, \boldsymbol{c})$ using the ASB algorithm, and then fix $\boldsymbol{c}$ and solve $(\boldsymbol{g}, \boldsymbol{B})$ using quasi-convex programming techniques. We iterative though these two stages until a local optimal solution is found.

3.2.2. Multiplexing Delay Sensitive Multimedia Traffic

We further consider the transmissions of delay sensitive multimedia traffic, where the typical stringent delay requirements do not permit the approach of using effective bandwidth theory, i.e., a large buffer size implies large average delay. It is well known that stored video, for example, can only be transmitted successfully (satisfying stringent delay and loss probability constraints) by allocating enough bandwidth to the video traffic streams.[29] Hence, this motivates the use of bufferless model to study the performance of statistical multiplexing video streams, i.e., only bandwidth allocation needs to be considered.

Here we adopt the approach in the work by Zhang et al[29] where we consider multiplexing delay sensitive multimedia traffic based on a bufferless model and the marginal distributions of the traffic. We use the well known Chernoff bound to estimate the loss probability. For each DSL user, we determine how much bandwidth will be allocated to the multimedia traffic streams together with the bandwidth and buffer allocation to the data traffic flows. We assume that the QoS requirements for multimedia and data traffic are different from user to user, i.e., a heterogeneous network scenario.

We will follow essentially the same notation in Section 3.2.1, with additional superscript d denoting the data traffic and superscript m denoting the multimedia traffic. Mathematically, we have the following problem formulation:

$$\begin{aligned}
\max_{(\boldsymbol{g}^d,\boldsymbol{g}^m,\boldsymbol{c}^d,\boldsymbol{c}^m,\boldsymbol{c},\boldsymbol{B})\geq 0} \quad & \sum_n \left(w_n^d a_n^d g_n^d + w_n^m a_n^m g_n^m\right) && (10) \\
s.t. \quad & g_n^d \nu_n \left(-\log \epsilon_n^d / B_n\right) \leq c_n^d, \forall n, \\
& g_n^m \cdot \text{user } n\text{'s bandwidth per multimedia flow} \leq c_n^m, \forall n, && (11) \\
& \text{user } n\text{'s multimedia traffic loss rate} \leq \epsilon_n^m, \forall n, \\
& c_n^d + c_n^m = c_n, \forall n, \\
& \sum_n B_n = B. \\
& \boldsymbol{c} \in \mathcal{C}.
\end{aligned}$$

In other words, we need to decide how many data and multimedia traffic to admit for each user $\left(\boldsymbol{g}^d, \boldsymbol{g}^m\right)$, what is the corresponding bandwidth allocation $\left(\boldsymbol{c}^d, \boldsymbol{c}^m, \boldsymbol{c}\right)$ and how to allocate the buffer space $(\boldsymbol{B})$. Again, we design a two-stage algorithm to iteratively solve for these variables until a local optimal solution to Problem (10) is found.

3.3. Distributed Joint Scheduling and Spectrum Management

Statistical multiplexing at the link level is performed over a sufficiently long time interval where it is reasonable to exploit the stationary "stochastic" characteristics of traffic flows. On the other hand, scheduling should be done over a smaller interval to efficiently utilize the dynamically changing instantaneous achievable rates. The upstream transmission shares a similar problem structure with the downstream transmission which is the total weighted rate maximization problem where the weights are adapted at each scheduling time instance. Hence, the ASB algorithm that lies at the core of the machinery in our framework drives the scheduling in both downstream and upstream transmission. Beside the timescale difference in operation, the main distinction in "upstream scheduling" from "downstream scheduling "is that distributed dynamic algorithms are needed to coordinate the users across the space and time and ensure network stability simultaneously, i.e., queues in the transients do not build up indefinitely. As compared to downstream transmission, a milder form of statistical multiplexing in the upstream transmission is used to exploit the economies of scale.

3.3.1. A stochastic network optimization approach

Our joint spectrum management and scheduling algorithm involving physical layer parameters is inspired by the seminal work on dynamic algorithms that utilize the back-pressure scheduling approach.[19,20] The back-pressure scheduler is optimal in the sense that it allows transmission at the maximum possible arrival rates into the network for which the queues at the various network nodes are still stable (in the Lyapunov sense), i.e., no indefinite buildup of queue size. The back-pressure algorithm and the ASB algorithm share a similar mathematical structure to the utility gradient maximization approach.[1]

The objective of the DSL upstream transmission is to determine the power allocation for each user such that the total expected utilities of all users under appropriate performance or fairness constraints is maximized. The stochastic network utility maximization is formulated as:

$$\begin{array}{ll} \text{maximize} & \sum_{k\in\mathcal{S}} \pi_k \sum_{n\in\mathcal{N}} U_{n,k}(\mathbf{P}^k) \\ \text{subject to} & \mathbf{R}_{avg} = \sum_{k\in\mathcal{S}} \pi_k \mathbf{R}(\mathbf{P}^k, k) \geq \lambda, \\ & \mathbf{P}^k \in \mathbf{\Omega},\ \forall k, \end{array} \tag{12}$$

where $\mathbf{\Omega}$ is the power constraint of a DSM algorithm that achieves the set of all feasible rate vectors. The system state $\pi_k, k \in \mathcal{S}$ represents one of the possible levels of channel conditions which is collectively affected by the number of users, topology, loop length and noise perturbation. The utility function is assumed to be a general (typically concave) function that is continuously increasing in its argument for each channel state k. Problem (12) can be reduced to the standard DSM scheduling if we let $U_{n,k}(\mathbf{P}^k) = w_n \log(1 + \mathsf{SIR}_n(\mathbf{P}^k))$, $|\mathcal{S}|= 1$ and λ to be the minimum average rate requirement. We particularize $U_{n,k}(\mathbf{P}^k)$ as $U_n(g(\mathbf{P}^k)) - P^{k,n}$ where $g(\mathbf{P}^k)$ is the achieved throughput and $P^{k,n}$ is the total average power of the nth user at state π_k. $U_n(g(\mathbf{P}^k))$ is a general utility function which is a measure of end users' satisfaction.[9] Problem (12) is inherently nonconvex, and thus is unlikely to be solved optimally at each scheduling interval. Also, it may be impractical to obtain accurate channel state information or implement a centralized algorithm to solve each optimization problem accurately at each scheduling interval in the DSL network. Hence, a distributed algorithm for scheduling different users is necessary. The suboptimal solutions obtained by this distributed approach can be suitably quantified in terms of the loss in optimality and network parameters. We first highlight the features of our proposed distributive dynamic algorithm before we give a detailed description of the algorithm in the following.

Scheduling of Real-time Traffic Flows. Scheduling is performed at discrete time slots (at a finer timescale than downstream transmission). We assume a single queue at the link layer for all users and queueing dynamics is governed by the equation $Q^n(t+1) = \max[Q^n(t) - R^n(t), 0] + A^n(t)$ where $Q^n(t)$ and $A^n(t)$ denote the amount of accumulated and new data arrival in the tth time slot respectively, and $R^n(t)$ is the data rate computed by the ASB algorithm. To capture the delay sensitivities of the real-time traffic in the above utility maximization framework, we incorporate the method of barrier function in optimization theory to take into account the deadline constraint. When a particular packet of a real-time flow nears its deadline expiration, that particular packet has a relatively higher priority than other packets in transmission through the DSL network. In this way, we are able to consider the trade-off between guaranteeing the service of real-time traffic flows with deadline constraints while providing certain fairness to the elastic data traffic or non-realtime traffic flows (captured by the utility functions).

Admission Control. Admission control is necessary to ensure that the optimization problem formulation in (12) is feasible. If the traffic types have large rate requirements that are infeasible, or the backlog in each user is not large enough, this may result in excessive packet dropping which leads to QoS degradation. To support all traffic types adequately, amplitude control is required to shape the flow intensities at each input of the network. The traditional approach to admission control requires an a priori traffic descriptor in terms of the parameters of a deterministic or stochastic traffic model to be declared explicitly. However, such an approach may easily lead to over-conservative resource provisioning and does not exploit fully the economy of scale. We adopt a distributed admission control scheme that exploits both the time axis and the amplitude axis at each scheduling interval. Our scheme may lead to slight overbooking in the short term, but it has a self correction ability in the sense that the long term average admitted user rate requirement is feasible. Furthermore, our admission control scheme also integrates nicely with existing higher layer protocol for amplitude control.

3.3.2. Distributed Dynamic Algorithm

We describe below a practical dynamic algorithm that stabilizes the individual queues and achieves an average optimal utility with an average power expenditure that is arbitrarily close to the minimum possible power for upstream transmission.

Admission control: Every time slot instant, for each nth queue, we allow the set of arrivals $A^n(t)$ into the queue whenever $Q^n(t) \leq V^n \beta^n$ where β^n is an adjustable threshold parameter on the queue size.

Dynamic spectrum allocation with ASB: 1) Every timeslot instant, the nth user estimates the current queue backlog $Q^n(t)$ and channel gain and allocate a power vector according to the following optimization:

$$\begin{array}{ll} \text{maximize} & Q^n(t)R^n(\mathbf{s}^n(t)) - V^n \sum_k s_k^n(t) + c_k^{n,p}(t) \log(d_k^{n,p}(t) - Q_k^{n,p}(t)/R^n(\mathbf{s}^n(t))) \\ \text{subject to} & \mathbf{s}^n(t) \in \Omega(P^n(t)), \end{array} \tag{13}$$

where Ω denotes the ASB achievable rate region which is parameterized by an average power constraint $P^n(t)$ of the modem (Unlike in the static ASB algorithm, $P^n(t)$ can be adjusted locally at each modem) and $Q^{n,p}(t)$ is the current packet remaining size. $d_{k+1}^{n,p}(t)$ and $Q_k^{n,p}(t)$ is the deadline and remaining bits of the current pth packet at the tth time slot respectively. The index k denotes the number of slots used to service the pth packet. V^n and $c_k^{n,p}(t)$ are arbitrary control parameters.

2) Update the deadline of the pth real-time packet at the kth round of bit-loading the pth packet $d_k^{n,p}(t)$ such that $d_{k+1}^{n,p}(t) = [d_k^{n,p}(t) - Q^{n,p}(t)/R^n(\mathbf{s}^{n,\star}(t))]^+$ where $\mathbf{s}^{n,\star}(t)$ is the solution to (13). If packet is non-real-time, we let $d^{n,p}(t) = \infty, \forall t$.

3) Update $c_k^{n,p}(t)$ such that $c_{k+1}^{n,p}(t) > c_k^{n,p}(t)$ and $c_k^{n,p}(t) \to \infty$ as k becomes large. After a packet is transmitted successfully, reset $k = 0$.

Statistical multiplexing at the upstream transmission can be controlled distributively through the parameter β^n. When all users have infinite amount of data to transmit at high-speed, a larger β^n for the nth user permits more data to be injected into the network (which in turn entails a larger $Q^n(t)$ in Problem (13), i.e., more weight is given to the nth user). It can be shown that the parameter V^n can be chosen such that the average power constraint can be pushed arbitrarily close to the minimum possible value with a corresponding increase in average delay.[20] This affects a natural tradeoff between average queueing delay and the average power constraint of the modem. More importantly, this tradeoff can be used to strike a balance between upstream transmission among users with different loop lengths. Such a delay based approach at the link layer is viewed as providing some form of implicit signaling to end-user applications about possible congestion level in the DSL network, and therefore preventing a vicious cycle of increasing power expenditure at the link layer to overcome interference. As shown in Fig. 2, the longer loop length will typically experience a reduced data rate R_1 as compared to R_2 of the shorter loop due primarily to the higher loop attenuation and the crosstalk from the shorter loop. Intuitively, the above dynamic algorithm computes the spectrum allocation for the shorter loop with a *minimum* possible average power constraint on the modem. Coupled with the fact that the ASB algorithm differentiates tone assignment according to loop length, R_1 will receive a boost in achievable data rate. Assuming that $R_1 = R_2$ and both users transmit packet of same sizes, the price to pay for the shorter loop is a correspondingly larger queueing delay experienced by the packets of the shorter loop.

In addition, as shown in Fig. 2, a natural integration with a higher layer in the protocol stack, e.g., transport layer such as the TCP Vegas or FAST TCP[27] (note that FAST TCP refers to a version of TCP congestion

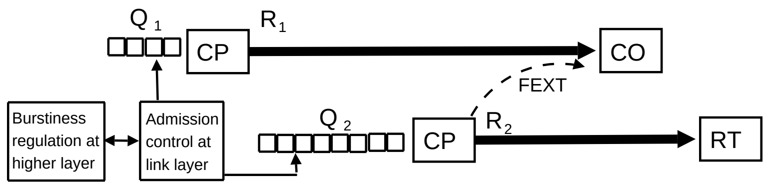


Figure 2. Distributed dynamic joint spectrum and scheduling using the ASB algorithm for a 2×2 DSL upstream scenario

control protocol, and has no relation to the FAST Copper project) provides a more robust solution in controlling the burstiness of traffic flows for upstream transmission. Particularly, FAST TCP is a delay based approach that controls TCP window size effectively with burstiness reduction.[27] The controllable parameter V^n at the link layer thus triggers an existing burstiness control for optimal throughput strategy in upstream transmission. At the same time, the control loop of the TCP window size adjustment is also shortened considerably and thus allows demand at the application layer to quickly match supply at the global link layer. By a suitable choice of $U_n(g(\mathbf{P}^k))$ in (12), the interaction between FAST TCP and the dynamic algorithm can be systematically analyzed. The access network flow control in the above dynamic algorithm is also necessary to cope with unresponsive flows or users which do not have burstiness regulation such as FAST TCP. Statistical multiplexing is thus achieved separately through the different layer coordination. The question of how to smoothly integrate flow control in layer 2 at access network and congestion control in layer 4 at the global level is a topic under current investigation.

4. SPACE

The "space" dimension consists of two types of problems: architectural decision problems, and topology design problems. The former considers with the division of functionalities between access and core networks, e.g., how large the access networks should be, where the various types of video servers should be placed, which network elements should be responsible for containing or reducing excessive bandwidth demand, etc. Here we will focus on the more tactical problems of topology design.

We consider two important topology design problems in FAST copper access networks. The first one is how to deploy the hybrid fiber/twisted pair architecture, taking into account the spectral interference mitigation. For the existing copper (twisted pair) network, after investigating the trade-off between the number of homes to be served by a remote terminal (the edge node with fiber/copper interface) and the distance from the remote terminal to the central office with efficient spectrum management as demonstrated in Section 2, we need to determine the location of the remote terminals and the fiber network connecting these remote terminals to the central office with the objective of either maximizing the number of homes to serve given the budget limitation or minimizing the total cost by serving all the homes. Note that, building the minimum cost fiber network to connect the remote terminals and the central office can be abstracted as the weighted Steiner Tree problem,[26] which is already NP-Complete.[15] Therefore, to provide a tractable solution to the whole problem, we propose a two-stage heuristic, i.e., first determine the location of remote terminals and then construct the fiber network. At the first stage, we can apply a greedy method to place the remote terminals as close to the central office as possible such that the total bandwidth requirements from all the home users can be satisfied. At the second stage, many existing approximation methods for Steiner Tree problem[26] can be employed.

The second major topology design problem in FAST project is to offer fast recovery for access networks after natural failures or malicious attacks. This is particularly important in the context of national and homeland security, since the access parts of the network infrastructure that aggregate increasing volumes of voice, image, video and data traffic from end users are usually the least protected. For economic reasons, it is in general acceptable to provide failure recovery for the (core) access network (from the remote terminals to the central office). On one hand, since the cost of deploying fiber (trenching or hanging along poles) is usually much higher than the cost of fiber itself, provisioning redundant bandwidth for automatic service restoration can be achieved without substantial additional investment. On the other hand, routing capability is very expensive,[2] thus only the central offices can be equipped with the routing capability, and the other terminals within the access network only have very limited switching capability (such as traffic aggregation using multiplexing and demultiplexing). Accordingly, the structure of the access network is a kind of "fat" tree, i.e., for an intermediate terminal node within an access network, the capacity/traffic of its upstream link is the aggregation of the capacity/traffic of all its downstream links. Therefore, to recover from its upstream link failure, the terminal has to relay the traffic from another terminal of the same or higher level. Such feature of the access network makes the problem of designing reliable access network different from that of designing reliable backbone (mesh) network.

Some interesting graph theory and optimization problems have been formulated from reliable access network design. One example is the *terminal Backup* problem, where we are given a graph with terminals (required vertices), Steiner (optional) vertices, and weighted edges, and the goal is to find the cheapest subgraph so that every terminal is connected to at least one other terminal (for backup purpose). Another interesting problem in topology design is to consider the connection between physical topology and the resulting type of crosstalk channel gains (which are topology-dependent even though they are time-invariant). Another subject to explore is the relationship between the topology deign and DSM algorithms.

Acknowledgement

We would like to thank the funding from the NSF CISE ITR program 2004-2008, collaboration with AT&T, and discussions with project co-PI John Cioff, Alexander Fraser, and their research groups.

REFERENCES

1. R. Agrawal and V. Subramanian. Optimality of certain channel aware scheduling policies. *Proc. of 2002 Allerton Conference on Communication, Control and Computing*, 2002.
2. M. Andrews and L. Zhang, "The access network design problem." in *FOCS*, 1998, pp. 40–59.
3. R. Cendrillon, G. Ginis, and M. Moonen. Improved linear crosstalk precompensation for downstream vdsl. *Proceedings of IEEE International Conference on Acoustics, Speed and Signal Processing (ICASSP)*, pages 1053–1056, 2004.
4. R. Cendrillon and M. Moonen. Iterative spectrum balancing for digital subscriber lines. *IEEE International Communications Conference (ICC)*, 2005.
5. R Cendrillon, W Yu, M Moonen, J Verlinden, and T Bostoen. Optimal multiuser spectrum balancing for digital subscriber lines. *IEEE Transactions on Communications*, 54(5):922– 933, 2006.
6. R. Cendrillon, J. Huang, M. Chiang, and M. Moonen. Autonomous spectrum balancing for digital subscriber lines. To be submitted to *IEEE Transactions on Signal Processing*.
7. Vincent M. K. Chan. Multiuser detection and spectrum balancing for digital subscriber lines. Master's thesis, Department of Electrical and Computer Engineering, Univeristy of Toronto, 2005.
8. HM Chaskar, U. Madhow, N.R. Center, and MA Boston. Statistical multiplexing and QoS provisioning for real-time traffic on wireless downstreams. *IEEE Journal on Selected Areas in Communications*, 19(2):347–354, 2001.
9. M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle. Layering as optimization decomposition: A mathematical theory of network architectures. To appear in *Proceedings of IEEE*, 2006.
10. S. T. Chung. *Transmission Schemes for Frequency Selective Gaussian Interference Channels*. PhD thesis, Stanford University, 2003.
11. G. Ginis and J. Cioffi. Vectored transmission for digital subscriber line systems. *IEEE Journal on Selected Areas of Communications*, 20(5):1085–1104, 2002.
12. J. Huang, R. Berry, and M. L. Honig. Distributed interference compensation in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(5):1074–1084, 2006.
13. J. Huang, R. Cendrillon, M. Chiang, and M. Moonen. Autonomous spectrum balancing (ASB) for digital subscriber lines. *IEEE ISIT*, 2006.
14. J. Huang, Z. Li, M. Chiang, and A. K. Katsaggelos. Joint rate control and scheduling for wireless upstream video streaming. *IEEE International Packet Video Workshop*, 2006.
15. R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. Plenum Press, 1972, pp. 85–103.

16. F.P. Kelly. Notes on effective bandwidths. *Stochastic Networks: Theory and Applications*, 4:141–168, 1996.
17. Z. Li, J. Huang, and A. K. Katsaggelos. Pricing based collaborative multi-user video streaming over power constrained wireless down link. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toulouse, France, 2006.
18. R. Lui and W. Yu. Low-complexity near-optimal spectrum balancing for digital subscriber lines. *IEEE International Conference on Communications (ICC)*, 2005.
19. M. J. Neely, E. Modiano and C. E. Rohrs. Dynamic power allocation and routing for time varying wireless networks. *IEEE Journal on Selected Areas in Communications*, 23(1):89–103, 2005.
20. M. J. Neely. Energy optimal control for time varying wireless networks. *IEEE Transactions on Information Theory*, 52(2):2915–2934, 2006.
21. F.L. Presti, Z.L. Zhang, J. Kurose, and D. Towsley. Source time scale and optimal buffer/bandwidth tradeoff for heterogeneous regulated traffic in a network node. *IEEE/ACM Transactions on Networking*, 7(4):490–501, 1999.
22. M. Reisslein, K.W. Ross, and S. Rajagopal. A framework for guaranteeing statistical QoS. *IEEE/ACM Transactions on Networking*, 10(1):27–42, 2002.
23. T. Starr, J. Cioffi, and P. Silverman. *Understanding digital Subscriber Line Technology.* Prentice Hall, 1999.
24. T. Starr, M. Sorbara, J. Cioffi, and P. Silverman, *DSL Advances,* Prentice Hall, 2003.
25. D. M. Topkis. *Supermodularity and Complementarity.* Princeton University Press, 1998.
26. V. V. Vazirani, *Approximation algorithms.* New York, NY, USA: Springer-Verlag New York, Inc., 2001.
27. D. X. Wei, C. Jin, S. H. Low and S. Hegde. FAST TCP: motivation, architecture, algorithms, performance. To appear in *IEEE/ACM Transactions on Networking*, 2007.
28. W. Yu, G. Ginis, and J. Cioffi. Distributed multiuser power control for digital subscriber lines. *IEEE Journal on Selected Areas in Communication*, 20(5):1105–1115, 2002.
29. Z.L. Zhang, J. Kurose, JD Salehi, and D. Towsley. Smoothing, statistical multiplexing, and call admission control for stored video. *IEEE Journal on Selected Areas in Communications*, 15(6):1148–1166, 1997.