The FAST Copper Project: Fundamental Research in Fiber/DSL Broadband Access

Yung Yi, Dahai Xu, Chee Wei Tan, Jianwei Huang, and Mung Chiang
Department of Electrical Engineering, Princeton University

Abstract

This is an overview of the ongoing FAST Copper project, which aims at at substantial improvements in rate, reach, reliability, and quality in copper-last-mile broadband access through fiber/DSL deployment, engineering innovations, and fundamental research. The project is funded by NSF, and is currently pursued jointly by Princeton University, Stanford University, and Fraser Research Lab. In this article, we outline the motivations, challenges, and research issues associated with the project, including connections with many branches of fundamental research, collaboration with industry partners, and emphasis on architectural decisions in broadband access networks. We also report some of the recent results by the Princeton team in each of the four dimensions: Frequency, Amplitude, Space, and Time.

Keywords: Broadband access networks, Digital Subscriber Loop, Graph theory, Interference channel, Multi-carrier systems, Optic fiber communications, Optimization, Scheduling, Spectrum management, Statistical multiplexing, Topology design.

I. Introduction to FAST Copper

FAST Copper is an on-going U.S. NSF funded project that started in 2004, and is currently pursued jointly 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 substantially improve the rate, reach, reliability, and quality in copper-last-mile broadband access to everyone with a phone line. This goal will be achieved through two threads of research: first, dynamic and joint optimization of resources in Frequency, Amplitude, Space, and Time (thus the name 'FAST') [1] and second, integration of communication, networking, computation, modeling, and distributed information management for architectural design of broadband access networks.

Access networks are often the rate-reach-reliability-quality bottleneck of end-to-end connections in wide area networks. Although the fiber-to-the-home solutions promise to provide broadband delivery, the *labor*

[1]FAST Copper project is completely different from the TCP FAST project at Caltech, which is a project that improves the transport layer protocol.

costs associated with fiber installation need to be divided over the number of customers served by the fiber. That last segment labor cost of fiber is the dominant limitation in broadband access.

In this project, 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 network architectures 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, with a fixed pipe-size supporting circuit-switched voice traffic. Therefore, there are two shifts of mentality that underlines the wide range of activities in FAST Copper. First, we model a bundled cable of twisted pairs as one aggregate *multi-user* communication system, where multiple users *compete* against and *cooperate* with each other in this system. We explicitly take into account the *crosstalk* effects (both near-end and far-end) that currently form the data rate bottleneck, and exploit potential *cooperation* in sharing limited resources. Second, today's traffic over broadband access, including voice, data, and video, are predominately supported by packet switched IP. We can exploit the *burstiness* of the application traffic through aggressive statistical multiplexing, controlled by admission control, traffic shaping, scheduling, and priority queuing mechanisms to ensure the desired tradeoff between the number of application flows supported and the Quality of Service (QoS) attainable. Intuitive as these basic ideas are, there is a surprising abundance of opportunities to improve over the existing DSL last mile through engineering implementation of these ideas.

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 (MIMO) communication environment of DSL across multiple layers in the protocol stack, which are summarized as follows:

- *Frequency*. 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) [1] allows maximum flexibility in allocating rates among competing flows, achieves much higher total data rates, and extends the reach of broadband access.

- *Time*. FAST Copper also leverages 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 in copper-based last mile access, 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.
- *Space*. In building robust and efficient broadband access networks, two issues are particularly important: how a hybrid fiber/twisted pair architecture can be designed to utilize the best of fiber-based and copper-based communication potentials, and how a logical topology can be designed to offer fast-recovery after natural failures or malicious attacks.
- *Amplitude*. We propose to install active 'amplitude control' mechanisms to shape the flow intensities at the edge to provide different QoS classes through admission control and 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.

The progress in the project comes from solving important problems in the fundamental research disciplines of information theory (multi-carrier interference channel), signal processing (multi-user transceiver design), optimization theory (non-convex and coupled problems), graph theory (survivable tree topology design), stochastic theory (processor sharing and queuing networks), distributed control (feedback control at different time-scales), and network protocol design (resource allocation and functionality allocation).

After surveying the brief summary of some of the latest developments and future issues of individual four dimensions, F, A, S, and T, we discuss the challenges of architectural design for broadband access networks by focusing on cooperation and coupling of four dimensions. We expect continuous progress to be made by all institutions in the project in the coming years.

II. Frequency: Spectrum Management

The main performance bottleneck of modern DSL networks is the *crosstalk* among different users (i.e., lines or modems) in the same binder (see Fig. 1). By deploying efficient DSM (Dynamic Spectrum Management) techniques and reducing excess crosstalk among users, a network operator can significantly increase the date rates and service reach of broadband access. This substantial increase at the physical layer affects the user-level performance and QoS at the upper layer, ultimately leading to a non-negligible change of economic values such as revenue increase and cost decrease from the perspective of service providers and subscribers, respectively.

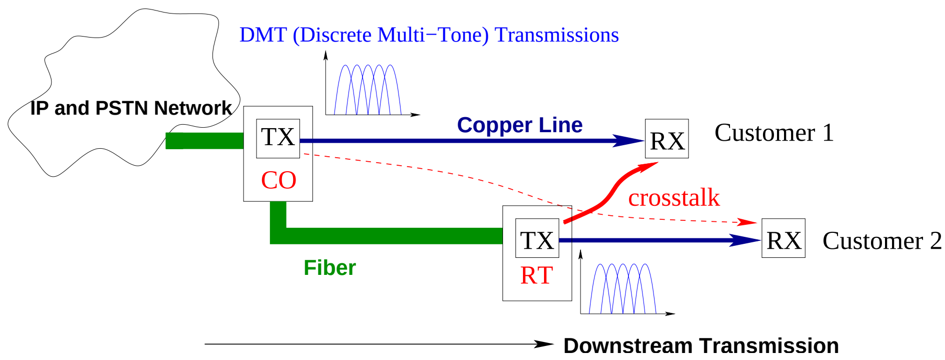


Fig. 1. The typical mixed CO/RT deployment (near-far scenario) for downstream transmission in an ADSL network. The CO (Central Office) is connected to the IP and PSTN Network via fiber; the RT (Remote Terminal) is connected to the CO via fiber. The CO and RT terminate at end customer homes through copper twisted-pair lines (telephone lines), where data rate is limited by crosstalk.

In a DSL system with multiple users and tones (i.e., frequency carriers), the objective of a DSM algorithm is to maximize the weighted sum rate for each user by suitably allocating the power spectrum densities (PSD), subject to the constraint that the total PSD per user is limited by a fixed amount. Conceptually, this problem is formulated by:

$$\max \sum_{\text{users},n} w_n R_n, \quad \text{subject to} \quad \textit{total power constraint}, \tag{1}$$

where w_n and R_n is the fixed weight and the rate for user n, respectively. Note that R_n is a *non-convex* function of PSDs of other users and tones, crosstalk interference, and channel gains. Thus, the non-convex optimization due to the coupling across lines and tones renders the DSM problem very challenging, and requires the global knowledge of the parameters, allowing only centralized algorithms.

The representative DSM algorithms in the literature to date (see [2], [3] for all references) include Iterative Water-filling (IW), Optimal Spectrum Balancing (OSB), Iterative Spectrum Balancing (ISB), and Autonomous Spectrum Balancing (ASB). The IW algorithm is a completely autonomous algorithm with a linear complexity in the number of users, In IW, each line selfishly maximizes its own data rate by water-filling over the noise and interference from other lines, but leading to a highly-suboptimal performance in the widely-encountered near-far scenarios. The OSB algorithm achieves the optimal performance by explicitly taking into account the damage done to the other lines when optimizing each line's spectra, but is a centralized algorithm with exponential complexity. Another centralized algorithm, ISB, improves over OSB by iterative optimization over the users, which, however, still requires quadratic complexity.

Recently, we have proposed the ASB algorithm, which is the first distributed, linear complexity algorithm that achieves near optimal performance. The key idea of ASB is introduction of a "reference line", a virtual line that represents a "typical" victim within the DSL system (e.g., the longest line). Then, instead of solving the original non-convex problem, each user tries to maximize a weighted sum rate and the reference line's rate, resulting in ease of computation. Intuitively, the reference line serves a static pricing term in each user's optimization problem to avoid purely selfish behavior, and eliminates the need of explicit message passing amongst users. The mathematics and functionality of ASB algorithm also play a key role in providing the physical layer rates to the upper-layer components, e.g., statistical multiplexing and admission control described in Section III.

III. Amplitude: Admission Control and Statistical Multiplexing

Statistical multiplexing has been widely studied in wire-line networks (e.g., [4] and the references therein), where the underlying link capacities are assumed to be fixed. In DSL network, however, the link capacities are closely coupled among users due to crosstalk interferences, and can be dynamically "shuffled" across users based on upper-layer traffic demands. Here, we aim at joint optimization of various system resources (e.g., rate and buffer) to squeeze as much traffic as possible into the network, subject to various flow QoS requirements. A typical QoS requirement is that the packet loss probability due to buffer overflow is upper-bounded by a fixed value less than 1. This probability is determined by the traffic characteristics (i.e., mean, peak, and variance) and the allocated resource to the traffic (i.e., bandwidth and buffer), where bandwidth allocation is provided by a DSL algorithm in Section II. In what follows, we describe an example of problem formulation and provide its solution. We refer the readers to [5] for details.

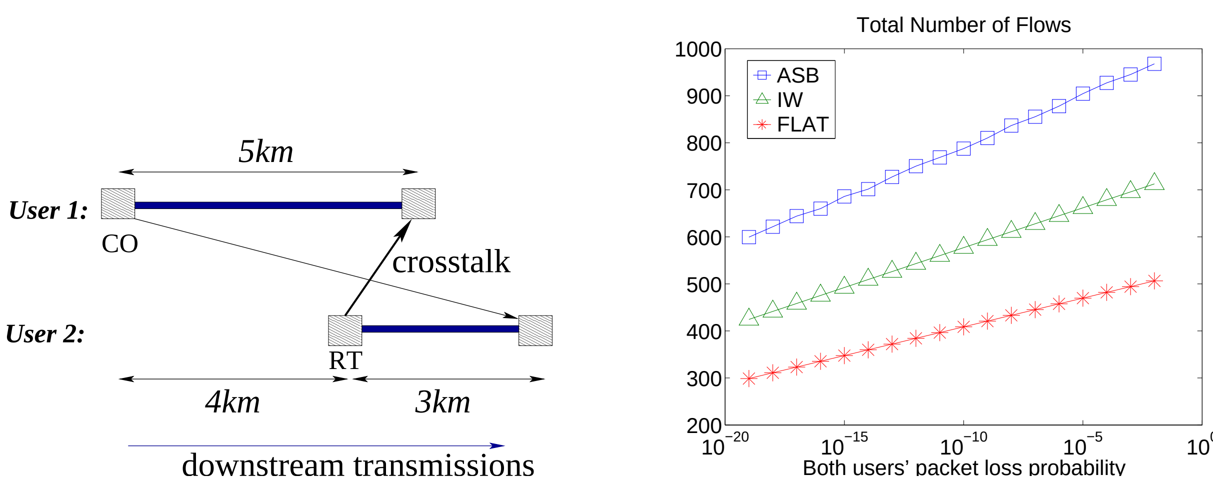
When the upper bound is very stringent and the available buffer space is large, the bandwidth requirements of users' data traffic can be estimated accurately using the well-established concept of *effective bandwidth* [6]. In the effective bandwidth theory, the required bandwidth (of user n) for the QoS guarantee ϵ_n and the buffer size B_n can be represented by a function $\nu_n(\epsilon_n, B_n)$ whose value is typically between the average and the peak rate. Note that each user n has its own separate function ν_n, since the statistics of input traffic could be different for each user n. This enables us to model a stochastic traffic with a constant traffic and perform admission control based on statistical multiplexing. The coupling of link capacities among users due to crosstalk interference enables us to recalculate the allocated rates to maximize the use of potential cooperation among users, such that the QoS for a new incoming flow together with those of existing flows are satisfied, if possible.

Then, our essential goal can be seen as a maximization of the admission region of the network, i.e., maximization of the number of flows of each user ($\boldsymbol{g} = \{g_n\}$) with their QoSs satisfied. We also need to allocate a separate queue for each user n with different buffer size B_n with $\sum_n B_n = B$, where B is the total system buffer size. Denote the rate of DSL link n (user n) as c_n, then the total effective bandwidth of user n's traffic should satisfy $g_n \nu_n(\epsilon_n, B_n) \leq c_n$. The vector $\boldsymbol{c} = \{c_n\}$ 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 a DSM algorithm in the frequency dimension. Mathematically, we have the following problem formulation:

$$\begin{aligned} \text{maximize} \quad & \sum_n g_n \\ \text{subject to} \quad & g_n \nu_n(\epsilon_n, B_n) \leq c_n, \forall n \\ & \sum_n B_n = B, \boldsymbol{c} \in \mathcal{C}, \\ \text{variables} \quad & \boldsymbol{g}, \boldsymbol{c}, \boldsymbol{B} \geq 0 \end{aligned} \tag{2}$$

where vector $\boldsymbol{B} = \{B_n\}$. Note that finding the global optimal solution of Problem (2) efficiently can be difficult since Problem (2) is typically *non-convex,* since $\mathcal{C}$ is a non-convex set, as discussed in Section II. However, a feasible solution can nevertheless be computed with a two-stage iterative Alternate Maximization (AM) algorithm [5], using our low-complexity, distributed ASB algorithm in the frequency dimension.

The first stage of the AM algorithm optimizes over the rate allocation assuming fixed buffer allocation, which can be simplified and solved using the ASB algorithm in Section II in a distributed fashion. The second stage optimizes the buffer allocation assuming fixed rate allocation, which turns out to be a quasi-concave maximization problem and can be solved using the bisection search. The algorithm alternates between these two stages until no further improvement can be found. Since each stage of the AM algorithm improves the system objective, which is upper-bounded, the AM algorithm always converges.



(a) Two line simulation topology (b) Comparison among different spectrum management algorithms

Fig. 2. Statistical multiplexing effect for different spectrum management algorithms

Fig. 2(a) shows the DSL copper two-line topology of our simulation. Fig. 2(b) shows the total number of flows that get admitted for different packet loss probabilities using three autonomous spectrum allocation algorithms. We compare the performance of the ASB algorithm with the Iterative Waterfilling (IW) and flat spectrum allocation. As shown in Fig. 2(b), the ASB algorithm outperforms the other two algorithms.

Note that we perform joint optimization of rates and buffers at the flow-level time-scale, i.e., when a new flow comes into the system. However, it might be possible to enlarge the admission region by scheduling the rate inside the feasible rate region more intelligently and more dynamically by running DSM algorithm at a faster time-scale. We discuss this issue in Section VI.

IV. Space: Access Topology Design

In this section, we discuss the space dimensional issue, focusing on the cost-efficient design of resilient access network, i.e., offering fast recovery for access networks after natural failures or malicious attacks, which is one of the main goals of the FAST Copper project. The access parts of the network infrastructure aggregate increasing volumes of voice, data, and video traffic from end users, but are usually the least protected, in contrast to ring-based metro networks and partial mesh backbone networks. This lack of protection, primarily due to the economic reason of high price per customer when provisioning survivable networks, makes the access network the bottleneck of end-to-end survivability. It is necessary to design survivable tree topologies, through the appropriate addition of a limited number of redundant links to the tree, that can provide the best survivability-cost tradeoff.

The structure of the access network is a “fat” tree rooted at CO (Central Office). In other words, 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 network.

Different optimization problems can be derived from the variations in the existence of fat tree, single level or multi-level tree, objective function (minimizing total cost to construct survivable access network or maximizing total revenue with limited budget) and the cost function of edge capacity (concave or uncapacitated fixed cost). Some of these models have been studied while many remain under-explored.

Table I summarizes the main results from recent works on the computational complexity (polynomial (**P**) or NP Complete (**NPC**)) of various problems related to survivable access network design.

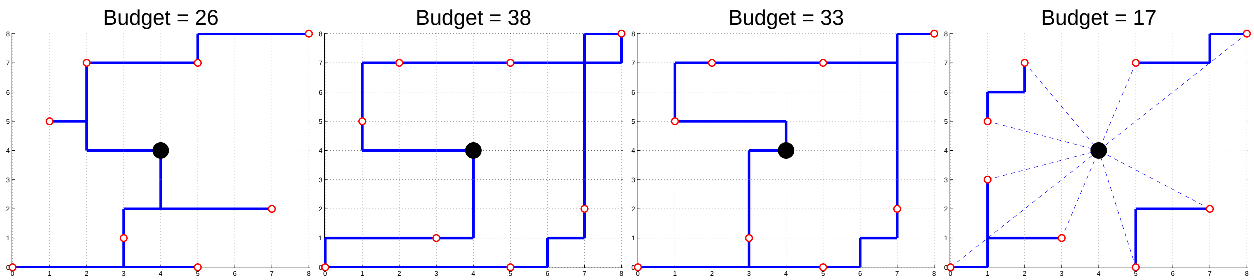
Particularly, we outline the following two new problems: (i) budget-constrained revenue maximization and (ii) provisioning survivability for existing single-level tree.

• *Budget-Constrained Revenue Maximization*. In this problem, uncapacitated fixed cost model is adopted and there is no existing tree. With a limited budget in network construction, we need to search for a subgraph to maximize the total revenue from providing survivability for partial remote terminals.

TABLE I

Computational Complexity of Survivable Access Network Design

Objective		Minimize Total Cost		Maximize Revenue with Limited Budget
Link Cost Model		Fixed	Concave	NPC [7]
Fat-Tree Exists	*Single-Level*	P [7]	NPC [8]	
	Multi-Level	NPC [7]		
No Existing Fat-Tree		NPC [9]		



(a) No Survivability (b) Full Survivability (c) Partial Survivability (d) Augmented Topology

Fig. 3. Access network design for a grid network

In the sample Manhattan-like grid network shown in Figs. 3(a)∼3(c), the empty circles are remote terminals and the solid circle at the center is central office. Each edge has 1 unit cost and the revenue by providing survivability (2-connectivity) to each remote terminal is 1 unit (different from unit edge cost). Fig. 3(a) shows the minimum cost (26 units) access network design without any survivability, and Fig. 3(b) shows the minimum cost (38 units) access network design with full survivability as well as the corresponding 8-unit revenue. In contrast, Fig. 3(c) shows the maximum revenue is 6 units when the budget is constrained by 33 units.

• *Provisioning Survivability for Existing Single-Level Tree.* In this problem, we search for the minimum cost incremental topology design to provide full survivability to all the remote terminals within an existing single-level access tree, which connects the central office and each remote terminal directly. Here, uncapacitated fixed cost model is used. To provide full survivability, we need to construct an augmented network where each terminal is connected to either one other terminal or the root. Such problem is abstracted as *Terminal Backup* problem, where we are given a graph with terminals (required vertices), Steiner (optional) vertices, and weighted edges. The goal is to find the cheapest subgraph so that every terminal is connected to at least one other terminal (for backup purpose). A sample minimum cost augmented graph is shown as bold lines in Fig. 3(d) where the existing access tree is shown with dotted lines.

Terminal Backup problem is shown to be equivalent to *Simplex Cover* problem [7], which involves

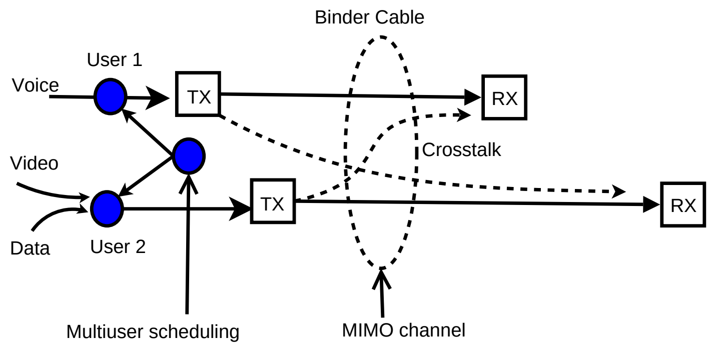


Fig. 4. Multiuser scheduling with two users in a MIMO channel with near-end crosstalk. Each user can send different number of traffic flows with different QoS requirements in the upstream and/or downstream of the binder cable.

finding an edge cover of some hypergraphs with certain nice properties. In terms of computational complexity, Simplex Cover is between regular Edge Cover (**P**) and Exact Cover by 3-SETS (**NPC**). In fact, we have proposed polynomial-time algorithms for both Terminal Backup and Simplex Cover [7], and demonstrated that the proposed approaches can solve the Simplex Cover problem efficiently even for large scale networks.

V. Time: Scheduling for Users and Flows

In the time dimension, we have the following two main issues: First, *which rate point* in the Pareto boundary of the feasible rate region, provided by the physical layer DSM algorithm, we choose and allocate to users, and divide (time-share) the allocated rate to individual flows inside each user, such that their QoSs are met. Second, *how often* we need to schedule the new rate for users and flows. In regard to the first issue, the scheduled points should clearly depend on the QoSs of flows in the system. The second issue corresponds to the time-scale of scheduling in the time dimension and DSM algorithm in the frequency dimension, which we discuss in Section VI. Fig. 4 shows the general framework of time-dimensional scheduling.

The whole flow-level QoSs inside each user comprise a user-level QoS, each of which is aggregated into an entire set of user-level QoSs. This user-level QoS set should be reflected in the rate allocation of multi user-level scheduling supported by a DSM algorithm, and the individual flow-level QoSs should be satisfied by a flow-level intra-user scheduling, given the rate by the multi-user scheduling. QoS can be specified in either deterministic or stochastic manner. An example of stochastic QoS is, as discussed in Section III, the upper-bounded packet loss probability or the delay violation probability. Examples of deterministic QoS parameters include real-time packet delay bounds and fairness requirements among flows. However, a deterministic approach may easily lead to over-conservative resource provisioning and does not exploit fully the economy of scale. Furthermore, the recent developments in multimedia processing technology on error resilience makes stochastic QoS requirements a more appropriate choice in the system design.

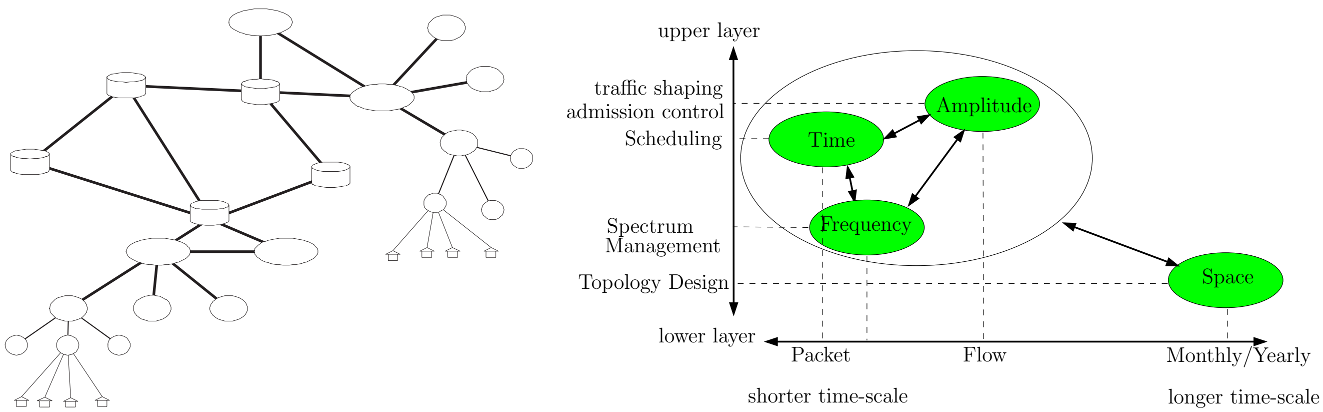
Recent trends in DSL networks have seen that voice, video, and data services are converging to one platform, which is predominantly supported by the packet-switched IP (e.g., voice/Internet/video Triple-Play product in the market). Thus, an important issue lies in the support of heterogeneous traffic flows having different QoSs. This means that intelligent scheduling algorithms are necessary to maximize the efficiency of system resource usage as well as satisfy various QoSs of different types of flows. We note that the percentage of real-time traffic in the DSL access network seems to be much higher than that in the normal commodity Internet. Generally, real-time flows require a certain set of QoS such as (stochastic or deterministic) packet delay and loss ratio. Thus, the injected volume of these flows to the system should be controlled by an appropriate admission control and flow intensity regulation (through policing or traffic shaping) at the amplitude dimension in Section III. On the other hand, non-realtime elastic data does not usually require a stringent QoS parameter, and operates on a best-effort basis. In this case, one possible "system-level" (not an individual flow-level) QoS might be fairness with the existing data flows inside the user and even with the flows of other users. However, depending on the products released by the service provider, there could be a "minimum throughput" guarantee service even for elastic data service, in which case admission control is again necessary.

One possible candidate of implementing differentiated service at user level is to select a priority function, such that widely different QoS requirements in terms of packet loss and instantaneous delay can be satisfied on a flow basis at each user. As an illustration, when a packet of a real-time flow for a particular user is close to its deadline expiration, the packet receives an increasingly higher priority than other packets belonging to other flows that are in transmission. Also, some real-time flows such as voice traffic may require low jitter to ensure good quality, thus a time-varying priority queueing discipline is necessary to ensure that different real-time and non-real-time flows with heterogeneous QoS requirements can be multiplexed at the MAC link layer. Designing appropriate priority functions for the priority queueing control mechanism based on traffic flow type and characteristics are currently under investigation.

VI. Architectural Issues in Broadband Access

A. Functionality Allocation

Offering data rate at 100 Mbps or above over twisted-pair already presents tremendous technical challenges, ranging from digital signal processing algorithms in the physical layer to the architecture/protocol design methodologies in the 'upper' layers. Even more challenging is the need to carefully investigate the coupling effects across multiple modules and across network elements, so that end-user experience over broadband access networks is enhanced, i.e., the access network *architecture* design.



(a) Horizontal decomposition

(b) Vertical decomposition and time-scale

Fig. 5. Functionality allocation

Architecture here refers to *functionality allocation*: which functional module and network element does what, and how they should be connected. Functionality allocation is often more influential, harder to change, and less quantitatively understood than any specific resource allocation scheme. Recent results in "Layering as Optimization Decomposition" surveyed in [10] have offered a useful conceptual framework and analytic tools for layered network architecture.

There are two types of functionality allocations:

- First, we use the term "horizontal decomposition" to refer to the geographic distribution of control into various network elements, e.g., in Fig. 5(a), from households to remote terminals and central offices (represented by circles), to larger central offices (represented by ovals), and to backbone acquisition, distribution, and video servers (represented by cylinders).
- Second, we use the term "vertical decomposition" to refer to the modularized design into a protocol stack, e.g., in Fig. 5(b). Ideally a fully integrated and joint design would be the best for performance optimization, but for many reasons such as evolvability and manageability, modularized design is necessary.

Horizontally and vertically decomposable modules have a variety of coupling relationships, depending on their time-scales and target service models. As an illustrative example, consider the functionality of error control. We can choose in vertically decomposable components from physical layer to application layer as well as horizontally decomposable components through either or both of a hop-by-hop or an end-to-end mechanism. In this article, we mainly discuss vertical decomposition through F, A, S, and T dimensions to understand how modules are connected, or even controlled so that different system performance can be achieved.

1) Coupling between Time and Frequency: Spectrum management in the frequency dimension defines the achievable rate points (i.e., rate region), whereas scheduling mechanism in the time dimension determines which rate point inside the rate region should be allocated to each user or flows, given the QoS. Ideally, time dimensional scheduling requires a packet-level granularity of control for rate allocation. For example, it is widely known that a scheduling algorithm for system "stability" (i.e., the number of packets at the buffer does not blow up) should allocate more rates to the user who has longer queue backlogs at every packet-level time-slot. However, non-convexity in the spectrum management renders such a fine-grained control of rates very difficult and even sometimes runs only in a sub-optimal manner. Thus, finding the (optimal) rate point required by the time dimensional scheduling at (fast) packet-level time-scale is almost impossible at frequency dimensional spectrum management. This explains the dependence of time dimension on frequency dimension, and raises issues about the *choice* of the algorithm as well as its time-scale.

2) Coupling between Amplitude and (Time, Frequency): The main problem in the amplitude dimension can be summarized as how to enlarge admission region, which is the set of arrivals with their QoS satisfied. The capability of scheduling in the time supported by spectrum management in the frequency actually determines admission region.

As an illustrative example, consider two scheduling algorithms (in the time dimension), π_1 and π_2 for the same set of QoS, where π_1 and π_2 are supported by two different DSM algorithms operating at the flow-level and packet-level timescales, respectively. Here, "time-scale" corresponds to the time to "solve" the rate allocation optimization of DSM algorithm in Section II. An example of π_1 is the rate scheduling in Section III, where new rates for lines are recalculated when a new flow comes into the system. We denote by $A(\pi)$ the admission region of a scheduling π.

Assuming that the time-scale of flow-level arrivals and departures is sufficiently slow, such that the stationary behavior can be seen during the inter-arrivals of flows, one potential tool of π_1 to schedule the rates and decide its admission region is the effective bandwidth [6], as discussed in Section III. On the other hand, the π_2 is able to schedule the arbitrary rates (which is inside the rate region) due to the support of the time-scale of the underlying DSM algorithm. Full support of a DSL algorithm in the frequency at the fast time-scale enables the π_2 to use inherent "opportunism", i.e., allocating more rates dynamically to the user whose transient situation requires prioritized service (e.g., the user with longer queue length) [11]. Then, π_2 can support the larger number of flows than π_1, but still can satisfy the same QoS as that in π_1.

This implies that $\mathcal{A}(\pi_1) < \mathcal{A}(\pi_2)$. Although π_1 with time-scale separation between time and frequency leads to better decomposability, it has to pay the price of decreasing system performance (i.e., decreasing admission region). This again illustrates the fundamental tradeoff between modularity and performance. It is important to have an efficient DSM algorithm which fills the gap, and to develop a joint statistical multiplexing, scheduling, and DSM algorithm to achieve better system performance.

For a given set of QoS parameters required by users and flows, admission region is decided by scheduling in the time dimension. Depending on the properties (e.g., time-scale and rate of convergence) of scheduling and spectrum managments, admission region is determined, and this region reflects the full set of flows that the system can support.

3) Coupling between Space and (Frequency, Time, Amplitude): Topology design (e.g., placement of various servers and schedulers) determines feasibility of control in the time and amplitude dimensions, since excessive propagation delay due to geographical distance may lead to instability of scheduling at the packet level and impossibility of admission control of short flows. Topology design also determines the crosstalk channel gains, which are the parameters to spectrum management algorithms in Frequency dimension and limit the best rate regions attainable. Then, as discussed earlier, time and amplitude dimension is closely coupled with frequency dimension.

B. Supoptimality and Time-Scale Issues

As discussed in the previous subsection, we ideally need the scheduling algorithm which operates and reacts to the change of system state (e.g., queue lengths and remaining deadline of backlogged packets) at the packet-level time-scale, which is typically on the order of milliseconds. However, due to the non-convexity of the weighted sum rate maximization problem in DSM algorithm and the requirement on distributed implementation in some cases, a large amount of time is often required to determine the global optimal rate allocation for all users. In Section II, we mentioned that even most of the existing practical DSM algorithms require a substantial amount of time to compute a feasible (suboptimal) solution in the Pareto region of the achievable rate region. In a couple of next paragraphs, we summarize the related issues and future research topics on how to analyze and improve.

First, we have *time-scale decision problem*: to use the most updated system state, but with suboptimal schedule, or to use the optimal schedule with out-dated system states. This decision problem depends on the physical-layer DSM algorithm, its behavior of approaching the optimal point, and finally the QoS parameters. The time-scale of rate scheduling algorithm affects the performance of flows differently, depending on the required QoS. A larger time-scale typically controls the performance metrics of the flows in the average sense (e.g., stability and long-term fairness), whereas a smaller time-scale operation has effect on the short-term QoS requirement of flows (e.g., delay and jitter). The choice of different time-scale for the rate computation in scheduling also allows a degree of freedom to decide the ‘degree’ of optimality at which resources are allocated. A suboptimal scheduling decision thus may be quantified through the loss in optimality in the objective function. Then, the important questions is : to what extent can we control the tradeoff between suboptimality gap and the requirement for less or even constant (with respect to the network factors) overhead?

The second topic, which is somewhat independent of the first topic, is on scalable and practical improvement of existing DSM algorithms considering *upper-layer performance.* The ASB algorithm that we have developed for achieving near-optimal performance with low-complexity helps much in improving the upper-layer QoS in some sense. However, existing DSM algorithms does not care about the upper-layer issues *explicitly,* which essentially leads to solving the optimization in (1) *from scratch.* However, in certain traffic conditions, the system state like queue length, which is used in the weight assignment of (1), typically does not change significantly over packet-level scheduling interval. Under this observation, one can devise an algorithm that computes a rate vector with lesser complexity that uses prior system-state information as opposed to solving a throughput maximization problem anew. A candidate approach is to use prediction algorithm that assigns a rate vector using previous rate vectors if the queue vector does not differ too much from the previous queue vector.

VII. Conclusion

As an on-going, multi-institutional project with collaboration across academia and industry, FAST Copper aims at enabling a truly ubiquitous and broadband wireline access network through fundamental research. There are two main types of research challenges in FAST Copper project. One is the array of problem formulations in the individual dimensions of Frequency, Amplitude, Space, and Time. These problems range from communication theoretic ones to graph theoretic ones, and from non-convex optimization to stochastic networks. The other is the need to quantify architectural principles in broadband access networks. This requires a characterization of horizontal and vertical decompositions for functionality allocation. In this overview paper, we have summarized the recent progress in 2005-2006 at Princeton University for the first type of challenges, and outlined the key issues involved in the second type that would benefit from many efforts from the research community.

Acknowledgment

We would like to acknowledge discussions with project co-PIs: John Cioffi and Alexander Fraser and their research groups, as well as co-authors of paper on FAST Copper: Elliot Anshelevich, Rapphael Cendrillon, and Marc Moonen.

References

[1] T. Starr, M. Sorbara, J. Cioffi, and P. Silverman, *DSL Advances*. Prentice Hall, 2003.

[2] 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.

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

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

[5] J. Huang, C.-W. Tan, M. Chiang, and R. Cendrillon, “Statistical Multiplexing over DSL Networks,” To appear in Proc. IEEE INFOCOM 2007.

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

[7] D. Xu, E. Anshelevich, and M. Chiang, “Simplex cover: Modeling, algorithms and networking applications,” in *submission to ICDCS 2007, http://www.princeton.edu/~dahaixu/pub/simplex/simplex.pdf*.

[8] M. Andrews and L. Zhang, “The access network design problem,” in *FOCS*, 1998, pp. 40–59.

[9] V. V. Vazirani, *Approximation algorithms*. New York, NY, USA: Springer-Verlag New York, Inc., 2001, vol. 11, no. 2.

[10] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, “Layering as optimization decomposition: A mathematical theory of network architectures,” *Proceedings of IEEE*, January 2007.

[11] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, “Scheduling in a queueing system with asynchronously varying service rates,” *Probability in the Engineering and Informational Sciences*, vol. 18, pp. 191–217, 2004.