7.0 Acknowlegements

K. Nichols would like to thank Van Jacobson for discussions clarifying CBQ, particularly with respect to constant bit rate service. M. Laubach would like to thank Steve Deering for inspiring the name for our variant of CBQ.

8.0 References

[1] S. Floyd and V. Jacobson, *Link-sharing and Resource Management Models for Packet Networks*, IEEE/ACM Transactions on Networking, Vol. 3 No. 4, pp. 365-386, August 1995

[2] See http://walkingdog.com for further information about IEEE 802.14.

[3] M. Laubach, *To Foster Residential Area BroadBand Internet Technology*, Connexions, February 1996 Vol 10, no. 2, p18-30.

[4] D. Sala and J.O. Limb, *A Protocol for Efficient Transfer of Data over Fiber/Cable Systems*, Proceedings of Infocom 1996, p 904.

[5] *LBNL's CBQ code*, available at http://www-nrg.ee.lbl.gov/floyd/cbq.html

[6] M. Laubach, *The UPSTREAMS Protocol for HFC Networks*, proposed to IEEE 802.14 Working Group, contribution number IEEE802.14-95/152R1, January, 1996.

FIGURE 12. Five CIR, Five BE, With a Preferential Weight Given to Overdraft in CIR vs BE

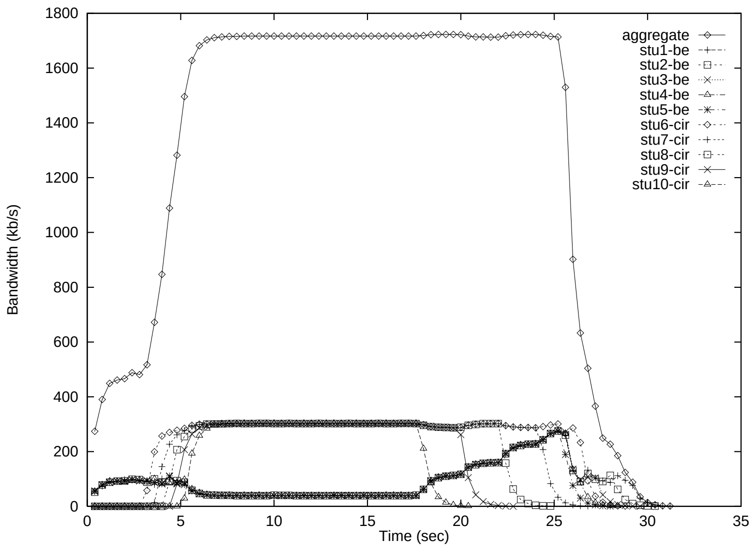
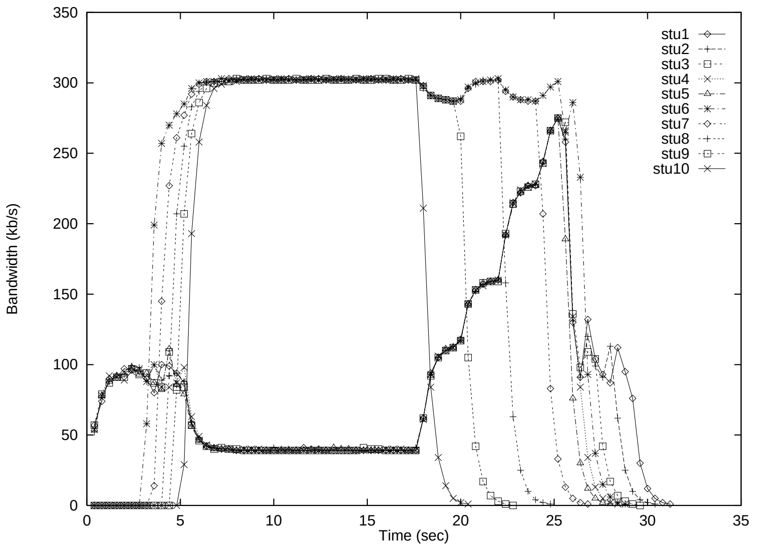


FIGURE 13. Figure 12 Without Aggregate Utilization



6.0 Future Work

Our prelminary results show that an SCBQ scheduler can deliver the QoS streams we require. Our future work will be in three categories: further algorithm refinement, further testing of SCBQ, and turning the simulation code into efficient runtime code. Further modifications to the allocation algorithm are needed to reduce CBR jitter and to allow properly sized bursts of cells from stations that are just becoming active. In addition, we need to do some further work on our hysteresis class to prevent STUs from leaving the list too soon when the channel is idle. There are a number of parameters in our system that need to be studied in simulation to gauge the sensitivity of the system to each and to properly set these values. We then need to run larger simulations with a variety of traffic types to study how well we can deliver the kind of performance that individual users will see. Along with this, we want to make the modifications to deliver multiple QoS streams to a single STU. Finally, our SCBQ simulation code has departed significantly from the original LBNL prototype code [5] and would need to be reworked for runtime efficiency.

FIGURE 10. Five CIR, five BE, with 4 CBR Sources

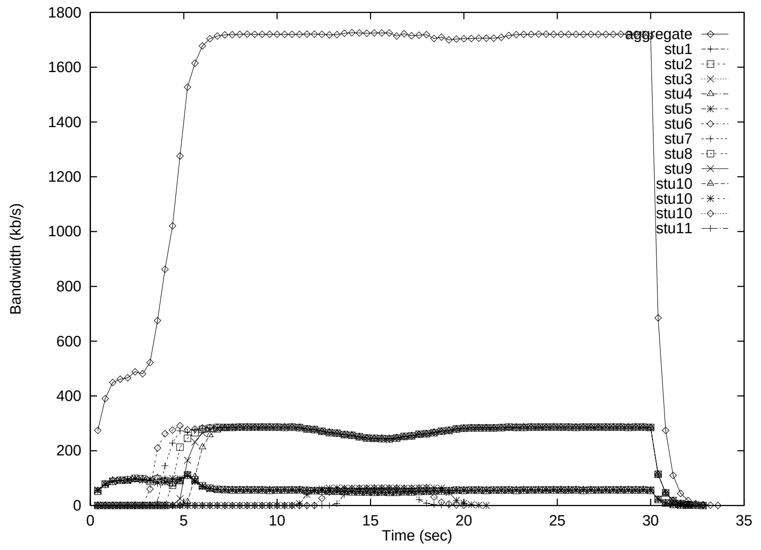
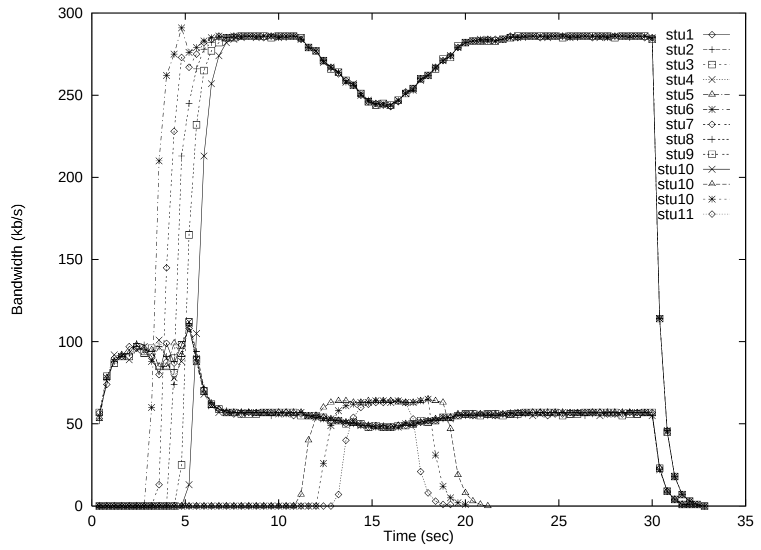


FIGURE 11. As in Figure 10 Without the Aggregate



In figures 12 and 13 we simulate with 5 BE STUs and 5 CIR STUs configured as in the previous example, but with CIR given a preferential weighting in the round robin allocation of excess bandwidth such that the BE class gets 2/3 the excess allocation of a CIR class. If the excess bandwidth were shared equally, we would have the previous situation of 295 Kbps for CIR and 51 Kbps for BE. However, in this scheme each CIR class could get an additional 177 Kbps and the BE class only 119 Kbps. Since 177 Kbps would put the CIR classes over their 300 Kbps upper bound, that leaves an additional 25 Kbps for the BE class to draw on. This means the 5 BE classes share (86+119+25) Kbps to get 46 Kbps each. This is reflected in figures 12 and 13.

FIGURE 8. Five BE STUs, Five CIR STUs, and a CBR STU

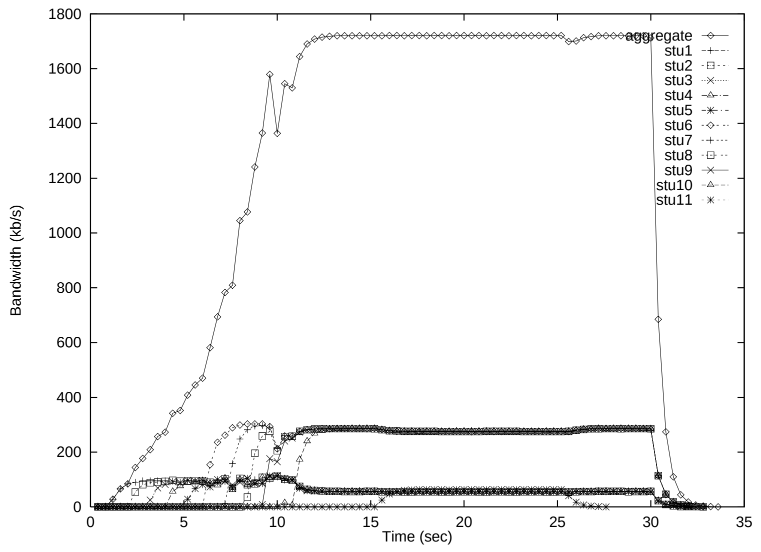
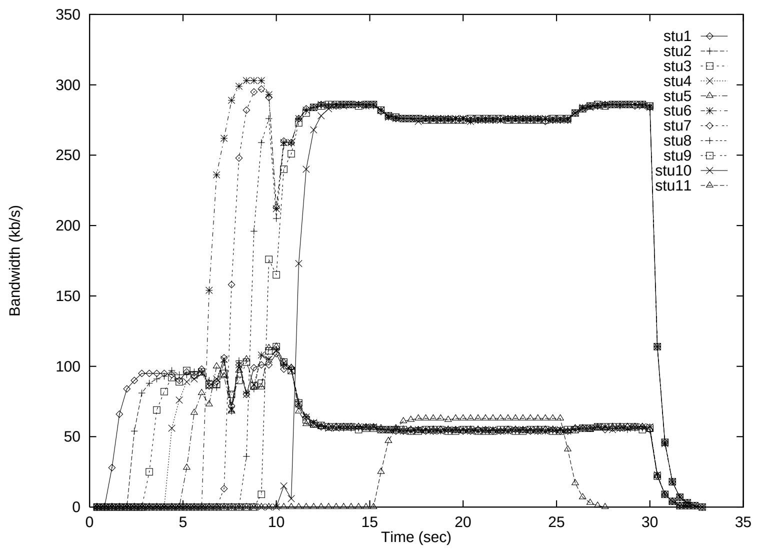


FIGURE 9. As in Figure 8 Without Aggregate Traffic



effort) being used. This leaves the rest to be shared equally between the six classes. Thus, each CIR STU should get 295 Kbps and the BE class gets 253 Kbps, to share equally among its 5 members, at about 51 Kbps. This appears to be happening. When the CBR source comes on, one-sixth of its bandwidth is taken from each CIR STU and one-sixth is taken from the BE class, spread equally among the 5 members. Although the plot is not at this level of resolution, the results appear to agree. In figures 10 and 11, the situation is the same before the CBR sources come on, but afterwards the total bandwidth for the six classes to share is 746 Kbps. This works out to about 250 Kbps per CIR STU and 42 Kps per BE STU. The actual numbers appear to differ slightly, but this could be because of the short hold time of the fourth CBR source (which was left off the plot).

FIGURE 6. Example of Figure 4 with a CBR (64 Kbps) Source Added

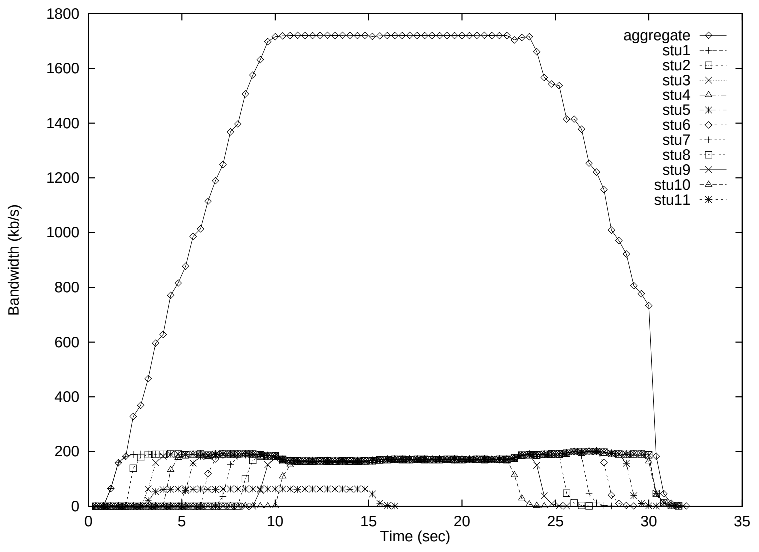
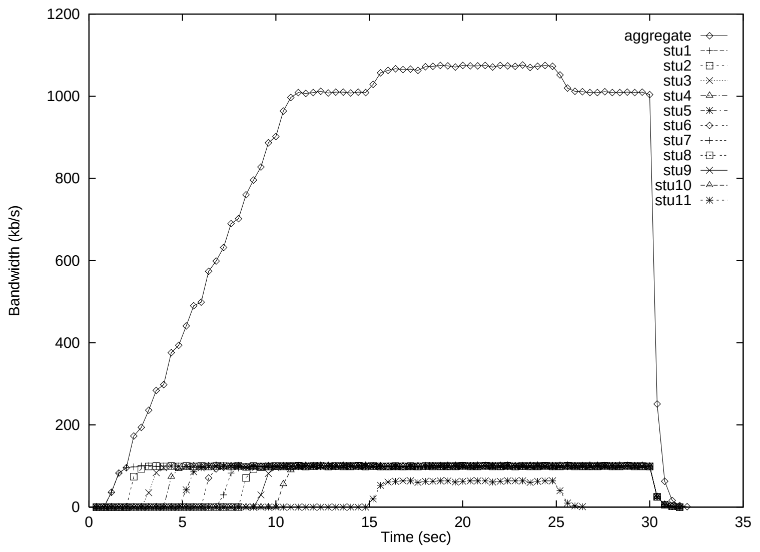


FIGURE 7. Example of Figure 6 With Reduced Upper Bound of (100 Kbps)Per CIR Source



Figures 10 and 11 show the results of the next experiment. Four CBR sources were added to the previous experiment (only 3 of these are plotted). Notice in figure 10 that we stay at the peak utilization the entire time that all sources are on. In figure 11 we can see that the overdraft allocations to both the CIR and BE are reduced equally during the time the CBR sources are active. We can also see that when the first CIR STU came on, it shot up to almost the entire 300 Kbps allotment before sharing with the next CIR source to come on. This is the appropriate behavior as we have configured it, but we'd like to be able to allocate the excess bandwith on an *unequal* basis, giving preference by subscription level.

First, we examine the sharing behavior In figure 8 and 9. Before the CBR source is turned on, there is 1.73 Mbps that has 657 Kbps of allocated bandwidth (the 128 Kbps of each CIR service plus the 5% of the link allocated to best

FIGURE 4. Ten Controlled Sources Sharing the Upstream Channel With Their Aggregate

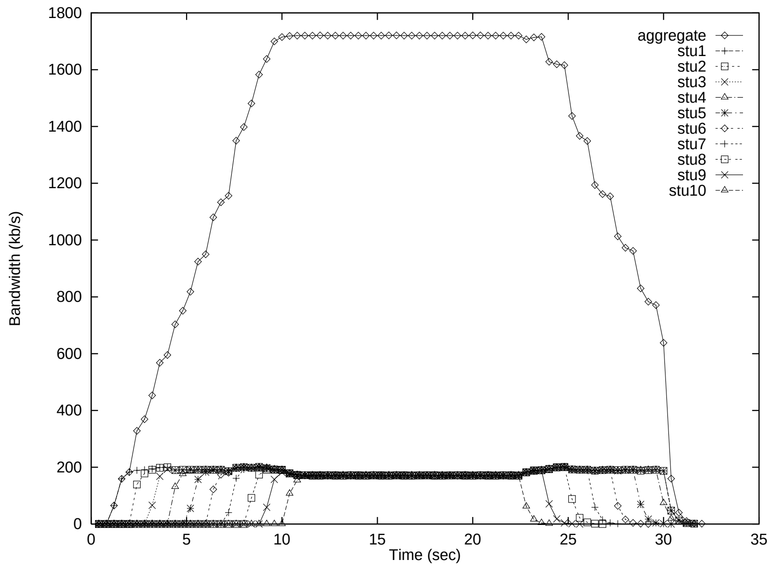
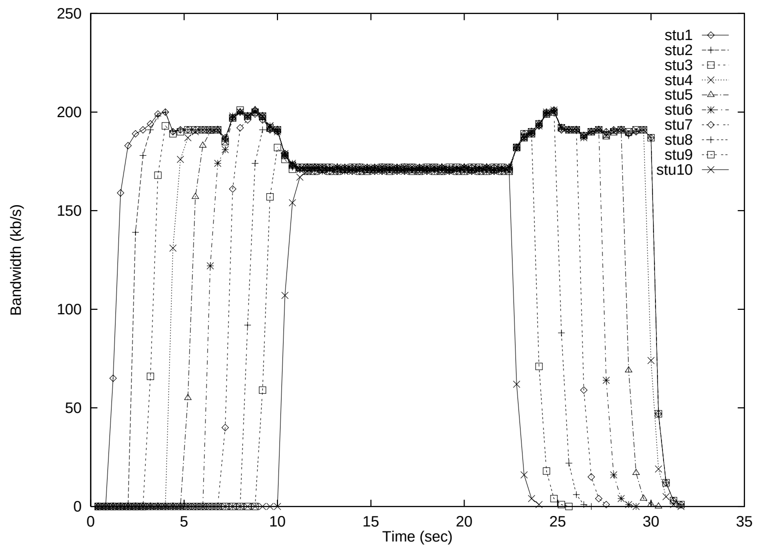


FIGURE 5. STU-Only Plot of the Same Experiment as in Figure 4.

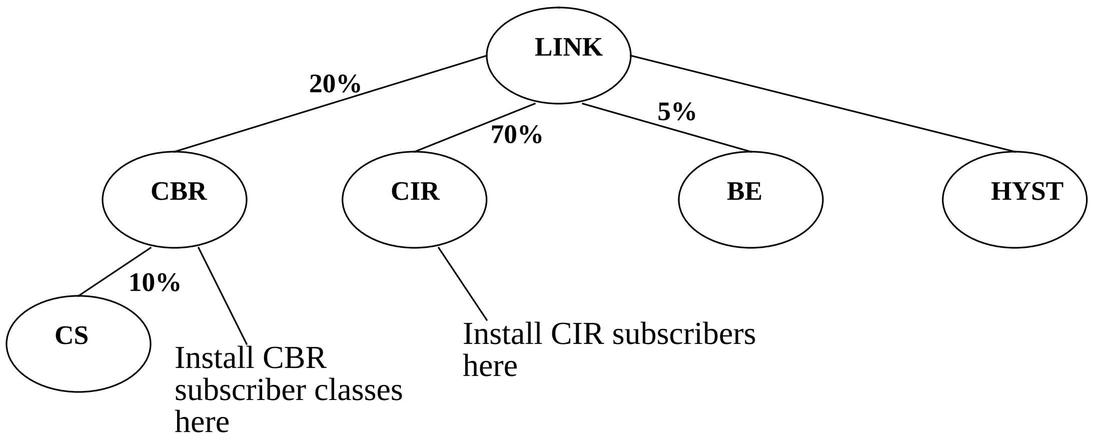


In the next experiments, we added best effort service. Recall, that this is a *true* best effort service where no minimum (other than an allotment of 1% of the link bandwidth for the entire class) rate is guaranteed per subscription flow, but no maximum is enforced either. Thus any spare bandwidth can be used by waiting traffic of this class. The sources being serviced at BE are sending at 96 Kbps and the CIR serviced sources are sending at 384 Kbps. The CIR service here is lower bounded at 128 Kbps and upper bounded at 300 Kbps. Each BE serviced source comes on at one second intervals from the simulation start, then the CIR serviced sources come in. A CBR source runs from 15 to 25 seconds. Figure 8 shows the experiment with the aggregate bandwidth value, which ramps up as additional bandwidth is used. In Figure 9 the same experiment is shown with just the STUs utilizations plotted so it can be seen that the BE service is reduced to accomodate the CBR source.

5.0 Results

SCBQ was implemented as the scheduler in a simulation model of an HFC UPSTREAMS system. The class hierarchy is headed by the link, under which there are four classes, CBR, CIR, BE, and HYST. CBR is priority 1, CIR and BE have the same priority, and HYST is the lowest priority possible. The CBR class includes both an administrative class that sends contention grants every ten slots (CS) and subscriber CBR, and is allocated a total of 20% of the bandwidth. The BE class has an aggregate allocation of 5% of the link, but may “overdraft” the unused link bandwidth. The subscriber CIR classes are installed under CIR and set up with both a guaranteed allocation and a maximum allocation. CIR itself cannot “overdraft” the link. In the experiments, constant bit rate sources were used to drive all the STUs. In most cases, we were driving at higher data rates in order to isolate scheduler effects.

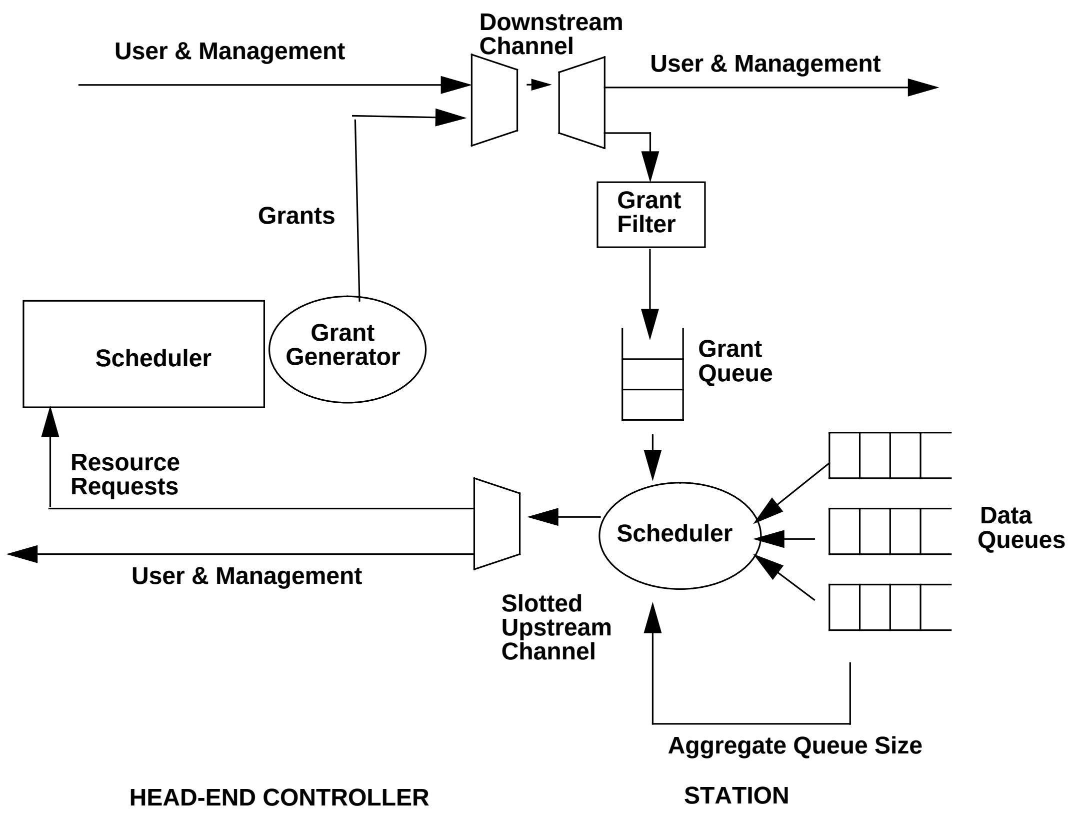
FIGURE 3. Class Hierarchy Use for the Simulation


LINK
20%
70%
5%
CBR
CIR
BE
HYST
10%
CS
Install CBR subscriber classes here
Install CIR subscribers here

In all the experiments, the CS class takes up 10% of link, reducing the available payload or data bandwidth to 1.73 Mbps. Each upstream ATM cell is sent in a time slot of duration 200 microseconds, giving a data payload of 1.92 Mbps. All simulations were run for 30 seconds of simulated time. The plots shown here are of the average instantaneous utilization. The first simulation example we’ll show is 10 STUs configured at *modem-quality* CIR and overdriving their allocation. Each source comes on at one second intervals into the simulation, runs at a bit-rate of 384 Kbps and turns off at one second intervals before the end of the simulation. The STUs have CIR service with a minimum guranteed rate of 28Kbps and a ceiling of 200 Kbps each. Since there is 1.73 Mbps available for sharing, each should get 173 Kbps, which is under the individual limiter value. In figure 4 we see that, indeed there is equal sharing of the channel. In figure 5 we show a close up of the STU utilizations only and can see that the STUs go after more of the channe bandwidth when there are fewer than 10 STUs, but that each is limited at 200 Kbps. Although we had only configured our CBR service in an *at most* rate delivery mode, we found that our average contention slot was 2.0 milliseconds as desired with a very small variance. Note that delays occur from the time a source starts or stops sending till it is heard by the HE due to round-trip delays in an HFC system. In additon, we can observe the backlogged STUs emptying their buffers.

In the next experiment, a constant bit rate subscriber stream of 64 Kbps, provisioned as a single cell every 6 milliseconds, was added. Note that when the CBR stream comes on at 3 seconds into the simulation it reduces the overdraft bandwidth available to the 10 CIR STUs. In figure 7 we show the results of repeating this experiment using a reduced upper bound for the CIR STUs of 100 Kbps. In this case, when the CBR source turns on it adds to the total link utilization since all 10 STUs are running at their maximum rate. In the experiment shown in figure 7, all 10 STUs were turned off at the end of 30 seconds.

FIGURE 2. Request and Grant Flow

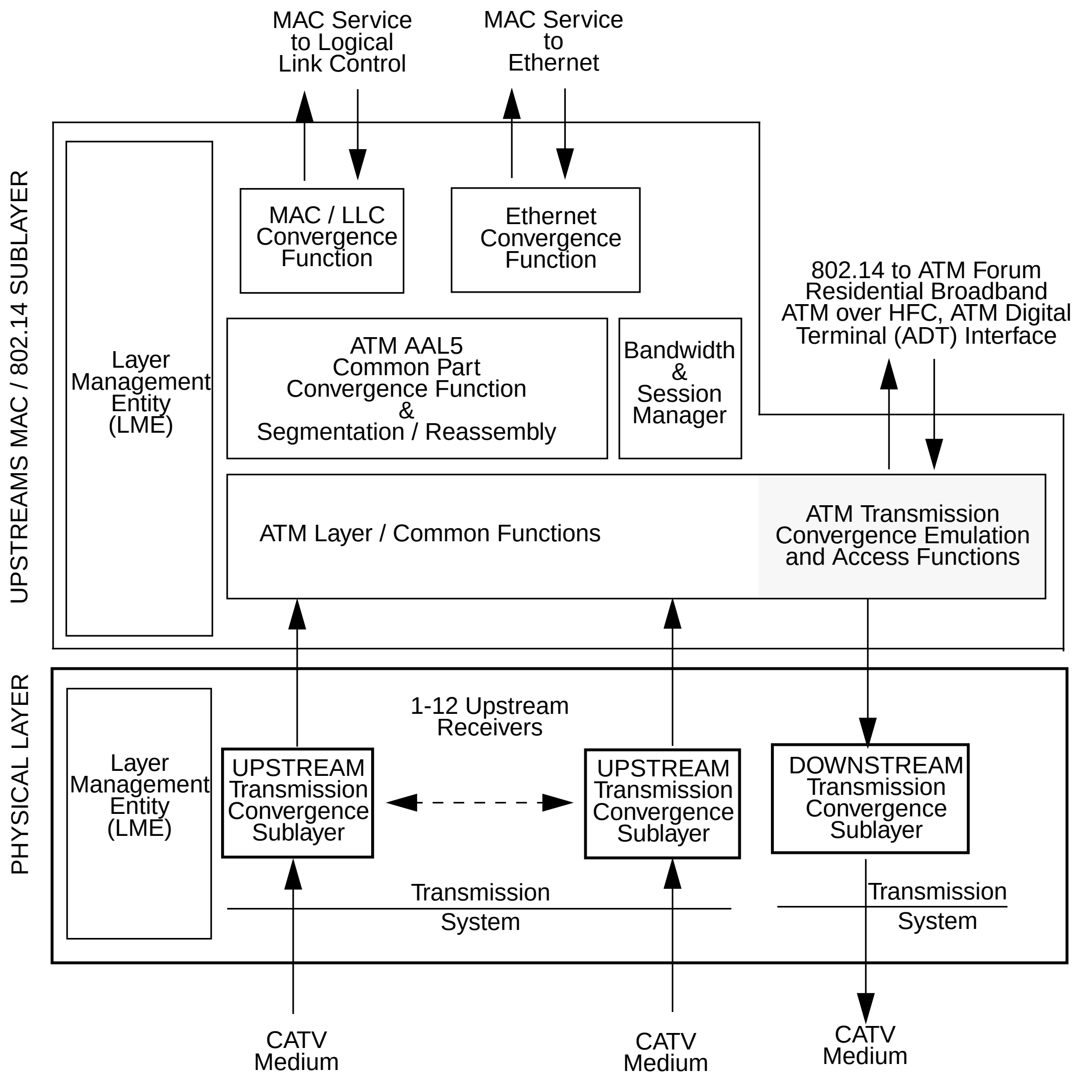


recently active users are likely to become active again in the near future, we want to apply some sort of hysteresis to the granting of upstream slots to stations. One way to implement this is by polling, but this introduces the potential for bandwidth waste when there are active stations requiring service. What is needed is a low-priority form of polling, where inactive stations are sent grants only when there is spare bandwidth and where STUs that continue to remain inactive could leave hysteresis eventually. As with the CBR service, we'd like a sheduler that can include these requirements in a integrated fashion.

The class-based queueing (CBQ) algorithm developed by Floyd and Jacobson [1] appeared to have the potential to meet our needs. We believed the concepts of classes, a class hierarchy, and priorities could provide our tiers of service. Still, CBQ was meant for packet-level scheduling and we needed to schedule grants based on undelineated requests. We realized that it would take some tuning to select the correct number of requests to schedule at a time as well as the fact that our scheduler works in an environment where it is necessary to schedule well in advance (due to round-trip delays) and to schedule some number of upstream cells at a time. To keep that number, and hence scheduler reaction time, small we need an algorithm that can be implemented in fairly short time and one that is extensible, as the number of active STUs on an cable can vary widely. CBQ looked like it could potentially work well under these circumstances.

We implemented a CBQ grant scheduler in our simulator. Since we are scheduling *shreds* of packets rather than full packets as in CBQ, we denote our algorithm *shredded* CBQ, or SCBQ. This scheduler has all the basic features we wanted, integrating all our requirements into a single scheduling mechanism, and has good sharing performance, as will be seen in the results. However, several aspects can be tuned further and we plan to do so in the future. We present our results so far as a promising proof-of-concept.

FIGURE 1. Block diagram of HFC controller for headend


MAC Service to Logical Link Control
MAC Service to Ethernet
UPSTREAMS MAC / 802.14 SUBLAYER
Layer Management Entity (LME)
MAC / LLC Convergence Function
Ethernet Convergence Function
802.14 to ATM Forum Residential Broadband ATM over HFC, ATM Digital Terminal (ADT) Interface
ATM AAL5 Common Part Convergence Function & Segmentation / Reassembly
Bandwidth & Session Manager
ATM Layer / Common Functions
ATM Transmission Convergence Emulation and Access Functions
PHYSICAL LAYER
1-12 Upstream Receivers
Layer Management Entity (LME)
UPSTREAM Transmission Convergence Sublayer
UPSTREAM Transmission Convergence Sublayer
DOWNSTREAM Transmission Convergence Sublayer
Transmission System
Transmission System
CATV Medium
CATV Medium
CATV Medium

bound, and no guarantee on jitters[1]; and a best effort (BE) category which receives available bandwidth only. The BE traffic may or may not be bounded, both as a class and by individual STU. Traffic models for the first category include low bit-rate alarm information, telephony, and conventional videoconferencing. Traffic models for the second category include classes of p*remium data service* like high-performance web surfing, work-at-home data traffic or Internet videoconferencing. Traffic models for the third category include electronic mail and text-based terminal sessions.

Constant bit rate is less problematic for a scheduler in a pre-allocated and fixed transmission unit size system since that rate can just be deliverd to the subscriber via direct grants at the appropriate intervals. However, we would like to implement a scheduler that handles the CBR and reactive data traffic in a monolithic fashion. The scheduling of the reactive traffic in the other two classes is complicated by the fact that, under normal circumstances, each new packet at the subscriber unit would have to contend to get a direct grant. A solution for very active STUs (e.g. doing an ftp or sending a video stream) is to piggyback the current number of enqueued data cells in each upstream cell so the scheduler gets an update of the required number of cells as part of the previously allocated bandwidth. For subscribers that are not active enough to keep the transmission cell queue full, this doesn't help. Relying on the observation that

1. CIR is similar to Available Bit Rate (ABR) when the Minimum Cell Rate (MCR) is non-zero.

create an upstream impairment challenge that requires a robust upstream RF modem that can cope with these impairment problems. Fourth, current IEEE standard efforts are to support a 50 mile cable length, which places approximately 400 microseconds of propagation delay between the cable head-end and the farthest subscriber station out on the network. Fifth, high utilization of the upstream bandwidth is necessary and can only be accomplished by sharing bandwidth among the stations and stations cannot directly communicate. The propagation delay of the cable together with techniques to reduce channel noise impairment introduces delays in the round-trip path between head-end and stations of several milliseconds. Meeting all of these requirements leads to a medium access control (MAC) layer where all stations are precisely ranged to a time-slotted channel that is shared among the subscriber stations under the control of the head-end. The precise time-slotting prevents stations from interfering with one another's transmissions and the head-end control allows for dynamic allocation of the bandwidth among the subscriber units to provide the level of service that a particular type of traffic requires and for which a particular subscriber has paid. Subscriber terminals communicate with the head-end controller to request access to the upstream channel and the head-end controller's scheduling algorithm allocates these time slots among the stations using the requests from the stations and knowledge about the level of service subscribed to at a particular station. This architecture has the advantage of reducing the complexity in the subscriber's unit to keep its cost down.[3, 4]

The raw downstream channel bandwidth is 30 Mbps with a usable data payload of 23.9 Mbps. The downstream channel is shared by all the subscriber units for both user and management traffic. The raw upstream channel bandwidth is expected to be about 2.56 Mbps with a usable data payload of 1.92 Mbps. However, multiple upstream channels can be employed on each cable, ultimately achieving a symmetric data rate if required. The standards call for supporting from 50-2000 stations on a single cable (see figure 1).

3.0 Overview of the UPSTREAMS protocol

Since stations can't communicate directly, it's not possible to use a DQDB or CSMA/CD type MAC layer. It is possible to use a slotted ALOHA technique, but the standards call for supporting connection-oriented traffic in addition to connectionless traffic and a high utilization of the upstream bandwidth is desired. The results presented in this paper are based on a Com21-developed protocol called UPSTREAMS: The Upstream Protocol for Sharing Transmission Resources among Entities using an ATM-based Messaging System [6]. This protocol has many features that are expected to be in the future IEEE 802.14 standard. The basic information unit that is sent in each upstream time slot is a 53 byte ATM cell augmented with 1 byte of management information, plus FEC and guardband bytes.

The head-end controls the upstream channel by issuing *grants* that specify the station(s) and type of messages that acan be sent in each upstream slot. Grants are may be issued to groups or all stations to sign on with the head-end (*invite grants*) or to make requests for bandwidth for reactive traffic needs (*contention grants*). Grants are issued directly to individual stations (*direct grants*) to send data in a particular time slot on the upstream channel. Since multiple stations can send in any invite or contention slot, random access algorithms are used for these slots to resolve contention when it occurs. Data is only sent in the direct grant slots. Multiple grants (up to 15) can be packed into a single downstream cell. A block diagram of the request and grant flow is shown in figure 2. General issues for the head-end scheduler, without considering quality of service, are: 1) grants must reach a station on the downstream in sufficient time for it to respond on the upstream channel, 2) don't schedule too far in advance (though system delays must be accounted for), and 3) avoid using contention when possible.

A scheduling algorithm is applied to the flow of resource requests using information about the type of subscription service of each traffic flow. Grants are then allocated to specific upstream cells based on this scheduler input. The head-end scheduler block in figure 3 implements our three-tiered service model while the scheduler at the station schedules based on simple priority.

In the following, we will use HE to denote head-end and STU to denote the station, or subscriber terminal unit.

4.0 Scheduling for QoS

Our QoS model has three categories of traffic: CBR, where each connection receives a constant bit rate delivered at minimum possible jitter; CIR, a *committed information rate* service with a minimum guaranteed rate, a maximum

Proceedings of ATM96, August 1996, San Francisco, CA

On Quality of Service in an ATM-based HFC Architecture

Kathleen M. Nichols and Mark Laubach
Com21, Inc.
1991 Landings Drive
Mountain View, CA USA 94043
+1 (415) 335-1724 / +1 (415) 254-5882
nichols@com21.com / laubach@com21.com
URL http://www.com21.com/

Abstract

This paper discusses the Hybrid Fiber-Coaxial cable TV environment as a data delivery service providing subscribers with specific levels of service. We overview HFC for data delivery and Com21's UPSTREAMS architecture for HFC data delivery using an ATM cell based format. We address providing quality of service over this shared medium at three basic levels: constant bit rate (CBR), a guaranteed rate class we denote as CIR (committed information rate), and a best effort (BE) class. To provide QoS between each flow, we employed the basic mechanisms of the Class-Based Queueing work of Floyd and Jacobson and adapted them to our architecture. We present our preliminary simulation results, showing the potential of this approach. Further algorithm tuning and studies are in progress.

1.0 Introduction

Cable modems are an evolving technology with the potential to deliver data bandwidth speeds far in excess of that provided by traditional twisted pair public telephone networks. Cable TV operators and Regional Bell Operating Companies (RBOCs) are installing or rebuilding existing all-coaxial cable plants into two-way Hybrid-Fiber Coaxial (HFC) plants and investigating offerings of a wide range of both data and interactive services which will be most attractive to their subscriber base. Initially, these services will only provide Internet access and access to major information services (e.g., CompuServe, AOL, and Prodigy), but services are expected to expand to include multi-player gaming and collaborative services such as voice and desktop video teleconferencing. To adequately support these evolving services and to smoothly integrate into future networks, we have developed an ATM-based architecture for these shared medium systems that supports tiers of service.

In the next section, we introduce some of the characteristics of the HFC environment and the challenges to data delivery. In section 3.0, we outline Com21's basic architecture, in section 4.0 we discuss the application of CBQ to our architecture, section 5.0 presents simulation results and we conclude with some discussion of our next directions.

2.0 Data in the HFC environment

The IEEE 802.14 Cable TV Media Access Control [2] and Physical Protocol Working Group and the ATM Forum's Residential Broadband Working Group, for ATM over HFC, are both working on standardization of broadband data delivery. Many companies are participating in the standards process, some of these with cable modems already announced or on the market. The IEEE 802.14 Working Group is chartered with providing a single MAC and multiple PHY standard for cable TV networks. 802.14 must support IEEE 802 layer services and must also be *ATM Friendly.*The expected data interface of choice in the home is a 10 Mbps ethernet and attached devices are expected to be those which support the TCP/IP protocol suite.

Implementation of two-way interactive services on HFC networks comes with a set of engineering problems. First, cable TV systems are inherently asymmetric in nature, with much more bandwidth available downstream than upstream. Second, the upstream facility is typically located in a sub- or low-split frequency region, typically from 5 MHz to 40 MHz, laden with many noise impairment sources. Third, the presence of impulse noise, ingress noise, common-mode noise, micro-reflections, group delay, and the effect of inexpensive *do it yourself* CATV home wiring