Bandwidth Scheduling for Multi-Channel Packet Cable Telephony

Nir Naaman and Raphael Rom

Department of Electrical Engineering
Technion - Israel Institute of Technology
Haifa 32000, Israel
Email: {mnir@techunix, rom@ee}.technion.ac.il

***Abstract*— Cable networks have evolved from offering broadcast services to providing high rate two-way data services. In the next step, cable operators intend to use Voice over IP (VoIP) to provide cable telephony services. In a cable network the users are connected to the headend through a cable modem. The headend is responsible for allocating upstream bandwidth to the various cable modems. Each cable modem has access to several upstream channels but can use only one upstream channel at any given time. The headend can direct a modem to switch from one upstream channel to another.**

We consider the problem of scheduling packet telephony calls in a cable network. We show that the scheduling problem is NP-hard even in the case where all the calls have the same characteristics. We then suggest several approximation algorithms for the problem and investigate their performance. We address the problem of maintaining the tolerated jitter when switching a modem from one channel to another and explore the effect of the tolerated jitter on the performance of the scheduling algorithms. We show that the ability to switch channels considerably improves the performance of the scheduling algorithms.

I. Introduction

Data over CATV has emerged as one of the leading technologies for delivering broadband services over the local loop. The existing infrastructure of Hybrid-Fiber-Coax (HFC), used by most modern CATV networks, enables cable operators to provide bi-directional high bandwidth data channels, at relatively low cost. Initially cable operators offered only best effort type of service by sharing the available bandwidth equally among all subscribers. In recent years, however, cable operators have been looking for ways to provide additional services such as telephony and video. In this paper we refer to the Data-Over-Cable Service Interface Specification (DOCSIS) standard [1], [2] which is the leading standard for data over CATV networks. We adopt the DOCSIS mechanism for supporting different Quality of Service (QoS) requirements and focus on the problem of scheduling flows with Constant Bit Rate (CBR) type of service; a problem which is left open by DOCSIS. Cable operators intend to use CBR flows to offer Voice over IP (VoIP) services that would compete with telephony services offered by telephone companies (see e.g., [3], [4] and [5] for more details).

CATV networks are characterized by a tree-and-branch topology. The headend or Cable Modem Termination System (CMTS), at the root of the tree, controls all traffic in the network. Subscribers of data services use a Cable Modem (CM) to connect to the CMTS. The available bandwidth is divided into channels. Downstream channels (CMTS to CMs) are used only by the CMTS. Upstream channels (CMs to CMTS) are shared by many subscribers (typically 500 to 2000). To share an upstream channel a TDMA MAC protocol with dynamic bandwidth allocation is implemented. In order to increase the overall bandwidth and improve noise immunity each modem can access multiple upstream channels. However, in order to reduce hardware and scheduling complexities a modem is usually restricted to use only one upstream channel for all its connections at any given time. The CMTS can instruct a CM to switch from the channel it is currently using to a differen channel.

DOCSIS based networks transfer Internet Protocol (IP) datagrams between the CMTS and the CMs. The CMTS is responsible for the scheduling of all transmissions in the upstream. Scheduling is done by dividing the upstream, in time, into a sequence of numbered mini-slots. A mini-slot typically carries 8-32 bytes. The CMTS and a CM establish a *service flow* between them; a service flow describes the type of connection between the CMTS and the CM and is identified by a service identifier (SID). A CM can support multiple active service flows simultaneously. In order to support different QoS demands DOCSIS defines several types of service flows. The main services are Unsolicited Grant Service (UGS) intended for CBR flows, Real-Time Polling Service (rtPS) intended for VBR flows, Non-Real-Time Polling Service (nrtPS) intended for non real-time VBR flows, and Best Effort service. We are interested in the Unsolicited Grant Service which is designed to support real-time service flows that generate fixed size data packets on a periodic basis. The main application of UGS flows is the delivery of VoIP packet telephony calls. When a user wishes to make a new VoIP call the CM tries to establish a UGS service flow with the CMTS. The CMTS decides whether to accept or reject the call; if the call is accepted the CMTS must allocate a fixed number of mini-slots to the CM at periodic intervals (see [5] for exact description). An active UGS service flow is characterize by three parameters: grant size (number of bytes to be allocated in each grant), grant interval (the delay between successive grants), and a tolerated grant jitter. Each flow may have a different set of parameters depending on factors such as the type of CODEC (Coder-Decoder) used, the bandwidth and buffering requirements of the application, and the number of active sessions (grants per interval) supported by the flow. The scheduling algorithm at the CMTS must allocate the available mini-slots to the different service flows while ensuring that each service flow is scheduled according to its specifications.

Previous work on scheduling real-time flows in DOCSIS compliant cable networks have been published. Scheduling of Variable Bit Rate MPEG video has been considered in [6], [7]. Simulations of simple scheduling algorithms for CBR flows have been presented in [8], [9]. The problem of scheduling CBR flows with different parameters on a single upstream channel is analyzed in [10]. In this paper we adopt the model of the scheduling problem presented by Wai Sum Lai in [11]. Lai considered the problem of assigning upstream channels and time slots to cable telephony calls and provided simulation results for several scheduling algorithms. Our work provides analytic worst case analysis of the problem.

Although we present the scheduling problem in the context of cable networks, similar problems may appear in other networks; for example, in satellite and fixed wireless networks that share many common characteristics of cable networks. It is interesting to note that algorithms similar to those we study have also been suggested for channel allocation in general packet radio service (GPRS) systems [12]. The scheduling problem in GPRS systems is different however since a mobile station is allowed to use several channels simultaneously and channel switching is not supported [13].

The rest of the paper is organized as follows. Section II provides a formal definition of the scheduling problem. In Section III we show that the problem is NP-hard even if all the calls have the same set of parameters. We describe several scheduling algorithms in Section IV and investigate their performance. In Section V we analyze the problem when jitter constraints are not present; we then extend the analysis to consider jitter constraints in Section VI. Section VII concludes the paper and suggests directions for further research.

II. Problem Definition

We consider the problem of scheduling packet telephony calls in a multi channel cable network. A call (UGS service flow) is characterized by three parameters

- Grant Size - S - the number of mini-slots that must be allocated to the call in each grant.
- Grant Interval - I - the nominal time between every two grants to the call.
- Grant Jitter - J - the tolerated delay of an actual grant from its nominal time.

The problem of scheduling calls with two or more different grant intervals is NP-hard even if the calls are to be scheduled on a single channel [10]. In this paper, we therefore restrict our attention to the case where all the calls have the same values of I, S and J. We divide the time axis of each upstream channel into frames of length I. Each call requires an allocation of a *time-slot* (of length S) in each frame. To describe a schedule it is therefore sufficient to describe the assignment of calls in a single frame in each channel. When all the calls are identical scheduling on a single channel is trivial; a call is accepted if there is a free time-slot in the frame and is blocked otherwise.

We assume there are m upstream channels and each channel has U time-slots per frame. Since all the calls of the same modem must be scheduled on a single upstream channel, each modem may have at most U calls simultaneously. The headend can direct a modem to switch from one upstream channel to another, in which case all the ongoing calls of the modem must be scheduled on the new channel. When a modem is switched to a different channel each of its ongoing calls must not be jittered by more than the tolerated jitter J.

An instance of the scheduling problem is a list of calls L. Each call has a starting (arrival) time, i.e., the time in which the modem tries to establish the call, and a duration. We distinguish between two cases

- Permanent calls - A call that has been scheduled lasts forever.
- Temporary call - Each call has a finite duration for which it must be scheduled. Once a call completes its duration it is removed. The duration of the call is unknown when the call is established.

In this paper we cover the case of permanent calls. Temporary calls are briefly discussed in subsection V-B (the complete analysis appears in [14]). Analysis of permanent calls may be justified by the fact that the duration of a call is typically several minutes while the time periods considered by the scheduler are typically only hundreds of milliseconds.

To evaluate the performance of a scheduling algorithm A we denote by $A(L)$ the number of calls A accepts from the list L, and by $OPT(L)$ the number of calls an optimal offline algorithm can accept. We let $R_A(L) \equiv A(L)/OPT(L)$ and define the performance ratio of algorithm A as

$$R_A \equiv \sup\{r \leq 1 : \ R_A(L) \geq r \text{ for all lists } L\} \qquad (1)$$

We concentrate on the online scheduling problem where calls are established over time. When a modem wants to establish a call it requests the CMTS to allocate the required bandwidth (a new time-slot). The CMTS must either accept the call and allocate the bandwidth or block (reject) the call. The scheduling decisions at the headend are made without any knowledge of calls that may be established in the future.

III. Complexity of the Scheduling Problem

We show that even the offline version of the scheduling problem is NP-hard. In the offline version the number of calls each modem has is known to the scheduling algorithm and all the calls are established at the same time. Note that in the offline problem channel switching is not required, hence jitter constraints are irrelevant and can be ignored. We define the decision version of the scheduling problem in the following way: given a list of calls L, can all the calls in L be scheduled using m channels?

Claim III.1: The decision version of the scheduling problem is strongly NP-complete.

Proof: We show a reduction from BIN PACKING (defined below) which is known to be strongly NP-complete [15].

BIN PASCKING:
INSTANCE: A finite set A of integers (items) $a_1, a_2, ..., a_n$, a bin capacity $B \in Z^+$ and a positive integer k.

QUESTION: Can A be partitioned into k disjoint subsets (bins) $S_1, ..., S_k$ such that $\sum_{j \in S_i} a_j \leq B$ for $i = 1, ..., k$?

As the scheduling problem is almost identical to bin packing, the reduction is straightforward. We construct a scheduling problem with $m = k$ channels, $U = B$ calls per channel, and n modems. We denote by M_i the number of calls of modem i and choose an instance in which $M_i = a_i$, $1 \leq i \leq n$. The problem of scheduling all the calls is now identical to solving bin packing. We associate between the modems scheduled in each channel and the items placed in each subset. This ensure that the answer to the scheduling problem is "yes" if and only if the answer to BIN PACKING is "yes". ■

Bin packing has been widely researched and many approximation algorithms have been developed for it (see [16] for a comprehensive survey). The similarity of the scheduling problem to bin packing suggests the application of bin packing algorithms to the scheduling problem.

IV. Online Scheduling Algorithms

We now consider the online version of the scheduling problem where calls are established over time. Since the problem is NP-hard we concentrate on approximation algorithms. A scheduling algorithm determines on which channel and in which time-slot to allocate each call. In this work we consider three algorithms. The algorithms specify the channel selection rules; in all three algorithm a new call is scheduled in the first available time-slot in the selected channel.

1) First Fit (FF) - Channels are numbered $1, ..., m$. The first call of a modem is scheduled on the first available channel. Subsequent calls are scheduled on the channel the modem is currently occupying. If the channel is full all the calls of the modem are switched to the first channel that can accommodate them. If there is no such channel the call is blocked.
2) Best Fit (BF) or Most Loaded - The first call of a modem is scheduled on the most loaded channel, i.e., the channel with the largest number of calls, (ties broken by lower index). Subsequent calls are scheduled on the channel the modem is currently occupying. If the channel is full all the calls of the modem are switched to the most loaded channel that can accommodate them. If there is no such channel the call is blocked.
3) Worst Fit (WF) or Least Loaded - Similar to Best Fit but chooses the least loaded channel.

V. The Scheduling Problem without Jitter Constraints

In this section we consider the case where there are no jitter constraints when switching calls from one channel to another. This situation occurs whenever $J \geq I$.

A. Permanent Calls

We first provide an upper bound on the performance ratio of any deterministic online algorithm.

Lemma 1: For the scheduling problem with permanent calls and no jitter constraints the performance ratio of any deterministic online algorithm A satisfies $R_A \leq \frac{2}{3} + \frac{1}{3U}$.

Proof: Consider first a family of algorithms that do not reject a call if it can be accepted. We choose $U \gg 1$ to be an even number and $m = 3$ channels. There are four modems $M_1 - M_4$ and each modem initially has $U/2$ calls. Regardless of the way the calls have been scheduled we know that one of the channels is full with the calls of two different modems; without loss of generality, let us assume it is channel 1 and the modems are M_1 and M_2. We now let one more call arrive to M_3. At this point we have the following configuration: M_1 and M_2 occupy channel 1 while M_3 and M_4 occupy two different channels (see Fig. 1.a). We now let M_1 and M_2 receive $U/2$ more calls each. These calls are all blocked by the algorithm, hence $A(L) = 2U + 1$. An optimal algorithm can accept $OPT(L) = 3U$ calls (see Fig. 1.b). The ratio in this example is $R_A(L) = \frac{2}{3} + \frac{1}{3U}$.

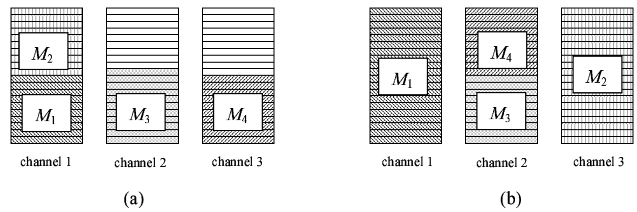

Fig. 1. Schedules of algorithm A (a) and an optimal algorithm (b) for the above example.

To complete the proof consider the case where the algorithm may reject a call that could be accepted. Suppose a call arrives to modem X and is rejected although it could be accepted. We repeat sending calls to modem X until it accepts a call, in which case we continue as in the above example, or until U calls have arrived to modem X. The calls that have been rejected could be accepted by an optimal algorithm so the performance ratio can only degrade. ■

Theorem 1: For algorithm $A \in$ {Worst-Fit, Best-Fit, First-Fit} with no jitter constraints the worst case performance ratio is $R_A = \frac{2}{3}$.

Proof: Lemma 1 provides an upper bound on the performance ratio of the algorithms (when $U \to \infty$). We now show that this is also the lower bound. To do so we evaluate the difference between the number of calls algorithm A accepts and the number of calls an optimal algorithm can accept. We assume A has scheduled several calls and consider the moment at which the first call is blocked by A. Suppose this call belongs to some modem X_1 that has x_1 ongoing calls at that time. We recognize the following properties:

1) The channel containing the calls of modem X_1 is full (otherwise the new call is accepted by A).
2) The content of any other channel is at least $U - x_1$ (otherwise modem X_1 is switched to a different channel and the new call is accepted).
3) An optimal algorithm can accept at most $U - x_1$ more calls to modem X_1.

Since X_1 can have at most U calls and one of its calls is blocked, the channel modem X_1 is occupying must have calls of at least one more modem. In a worst case example there is exactly one more modem that would have blocked calls in this channel. The reason is that, due to Property 2, if $x_1 \leq \frac{U}{3}$ the content of all the channels is at least $\frac{2U}{3}$, hence the performance ratio is at least $\frac{2}{3}$. Assume the next modem to receive a call which is blocked is modem X_2 and it has x_2 accepted calls. Clearly the same properties we listed above must also hold for X_2. Let $x = \min\{x_1, x_2\}$. In the worst case X_1 and X_2 occupy the same channel, which means that the number of calls accepted by algorithm A and an optimal algorithm satisfy

$$\begin{aligned} A(L) &\geq U + (m-1)(U-x) \\ OPT(L) &\leq \min\{mU,\ U + (m-1)(U-x) + 2(U-x)\} \end{aligned} \tag{2}$$

The performance ratio satisfies the following:

$$R_A \geq \frac{U + (m-1)(U-x)}{\min\{mU,\ U + (m+1)(U-x)\}} \tag{3}$$

The minimum possible value of the expression in (3) is $R_A = \frac{2}{3}$; it is obtained when $x = \frac{U}{2}$ and $m = 3$. This provides the necessary lower bound on the performance ratio of the algorithms. ■

1) Permanent Calls without Channel Switching: To evaluate the benefits of channel switching we now analyze the scheduling problem under the assumption that channel switching is not allowed, i.e., a modem is assigned a channel when the first call is established and all subsequent calls must be assigned to that channel.

A worst case example for algorithms FF and BF is straightforward. The first U calls are for different modems and they are all assigned to the first channel. In the next stage each modem receives $U - 1$ additional calls. Since the channel is full and channel switching is not allowed all the calls are blocked. We therefore have $FF(L) = BF(L) = U$ and $OPT(L) = \min\{U^2, mU\}$; assuming $U \geq m$ the performance ratio is $R_{FF} = R_{BF} = \frac{1}{m}$. This is the worst possible performance ratio since at least one channel is full before calls are blocked.

The WF algorithm performs better than BF and FF when channel switching is not allowed. However, we now show that the performance ratio is monotonically decreasing with U and therefore tends to $\frac{1}{m}$ as U increases. Consider the following example: in the first stage $k \cdot m$ calls for different modems arrive; as a result each channel is assigned k calls. In the next stage $U - k$ calls of a single modem arrive and are allocated in the first channel; at this point the first channel is full. We now have $U - 1$ calls arriving to each modem in the first channel; since the channel is full all these calls are blocked. The algorithm accepts $WF(L) = U + (m-1)k$ calls while an optimal algorithm can accept $OPT(L) = \min\{(k+1)U + (m-1)k,\ mU\}$. For simplicity we can choose $k = m$ and $U > m^2$ in which case the ratio is

$$R_{WF} = \frac{U + (m-1)m}{mU} = \frac{1}{m} + \frac{m-1}{U} \tag{4}$$

As U increases the performance ratio converges to $\frac{1}{m}$.

¿From the above examples we conclude that channel switching considerably improves the worst case performance ratio of the scheduling algorithms. When channel switching is disabled the performance ratio is monotonically decreasing with U and m; when channel switching is enabled the performance ratio is constant.

B. Temporary Calls

We now briefly discuss the case where calls begin and end over time. The duration of a call is unknown to the scheduling algorithm when the call is established. We assume that a call that has been accepted may not be interrupted.

When trying to evaluate the performance ratio for temporary calls we found that the performance ratio of any deterministic algorithm can be made arbitrary small. It is always possible to create a situation where the algorithm blocks a call which is accepted by the optimal algorithm. With temporary calls we can let this call end, and let a new calls with the same parameters immediately begin. If we repeat this process the optimal algorithm keeps accepting calls while the number of calls our algorithm accepts remains constant; hence the performance ratio tends to zero.

To overcome this problem we changed the definition of the performance ratio and defined it as the ratio between the total time of all the calls algorithms A and OPT accept. To get meaningful results we also had to set upper and lower limits on the duration of each call. The analysis is presented in [14]; due to size constraints it is omitted from this paper.

VI. The Scheduling Problem with Jitter Constraints

A. Scheduling using Jitter Windows

In this section we evaluate the use of jitter windows as a means for maintaining jitter constraints when switching a modem from one channel to another. The use of jitter windows was proposed in [11]. According to this proposal a frame containing U time-slots is divided into W windows such that the length of a window is less than the tolerated jitter of the calls, i.e., $\frac{I}{W} \leq J$. Hence, a call can be moved freely within its jitter window without violating jitter constraints. Jitter windows provide a simple way to ensure that when a modem is switched form one channel to another the jitter constraint of every call is not violated. It is important to note, however, that jitter windows add restrictions to the scheduling algorithm which are not imposed by the scheduling rules.

For simplicity we assume that U is a multiple of W; hence each window can hold up to U/W calls. A modem may use only one channel and within this channel it may have calls in different jitter windows. When a modem is switched to a different channel each and every call in the original channel must be allocated in the same jitter window in the new channel.

We first establish a lower bound on the worst case performance ratio of any algorithm that allows channel switching.

Lemma 2: Let A be an algorithm that allows channel switching and uses jitter windows. If A does not block calls unnecessarily the worst case performance ratio of A satisfies

$$R_A \geq \frac{1}{W} \quad \text{for every } W \geq 2. \tag{5}$$

Proof: We assume nothing about the way A schedules the calls. Our only assumption is that when a modem tries to establish a call the call is accepted in two cases: 1) The current channel the modem is occupying has a free time-slot, or 2) The call can be accepted if the modem is switched to another channel.

Consider a call that arrives to modem X and is blocked by A. Suppose modem X has x ongoing calls in jitter window W_j at that time. We recognize the following properties:

1) The channel containing the calls of modem X is full.
2) The content of W_j in all the other channels is at least $\frac{U}{W} - x$.
3) An optimal algorithm can accept at most $U - x$ more calls to modem X.

Algorithm A may block calls of other modems in the channel modem X is occupying. Let k be the number of modems with such blocked calls. Note that if a jitter window has two modems with blocked calls the content of the same jitter window in any other channel is at least $\frac{U}{2W}$. It follows that when more than one jitter window is occupied by two modems that have blocked calls, the performance ratio is already more than $\frac{1}{W}$. We may therefore assume that one jitter window, say W_1, has $1 \leq k_1 \leq \frac{U}{W}$ modems with blocked calls and in $0 \leq k_2 \leq W - 1$ other windows there is one modem with blocked calls. As a result, the number of calls accepted by A and OPT satisfy

$$\begin{aligned} A(L) &\geq U + (m-1)\left(\tfrac{(k_1-1)U}{k_1 W} + k_2\right) \\ OPT(L) &\leq min\{mU,\ A + (k_1 + k_2 - 1)U\} \end{aligned} \tag{6}$$

Simple analysis of the ratio $R_A = \frac{A(L)}{OPT(L)}$ shows that the worst case performance ratio satisfies $R_A > \frac{1}{W}$ for any set of values of U, m, k_1, and k_2. It converges to $\frac{1}{W}$ as $U \to \infty$ in two cases: either $k_1 = 1$ and $k_2 = W - 1$ or $k_1 = \frac{U}{W}$ and $k_2 = 0$.

The above result holds for any algorithm regardless of the way the calls are scheduled. ■

Let us now consider the Worst-Fit, Best-Fit and First-Fit algorithms. We first define the algorithms for the case of jitter windows.

Worst-Fit (Best-Fit) with Jitter Windows: The algorithm assigns the first call of a modem to the least (most) loaded channel and within that channel the call is assigned to the least (most) loaded jitter window. If a modem already has an ongoing call a new call is assigned to the least (most) loaded jitter window in the channel the modem is occupying. If that channel is full the algorithm tries to switch all the calls of the modem to a different channel. If there are several channels that can accommodate the calls of the modem, the least (most) loaded channel is selected.

First-Fit with Jitter Windows: The algorithm assigns the first call of a modem to the first available channel and within that channel the call is assign to the first available jitter window. If a modem already has an ongoing call a new call is assigned to the first available jitter window in the channel the modem is occupying. If that channel is full the algorithm tries to switch all the calls of the modem to a different channel. It examines the channel according to their order and select the first that can accommodate the calls of the modem.

Theorem 2: For algorithm $A \in$ {Worst-Fit, Best-Fit, First-Fit} using jitter windows the worst case performance ratio is $R_A = \frac{1}{W}$ for every $W \geq 2$.

Proof: Lemma 2 provides a lower bound on the performance ratio of the algorithms. We present an example for each algorithm that proves the upper bound.

Worst case example for Worst-Fit

We create a situation where the first channel is full such that each jitter window is fully occupied (has $\frac{U}{W}$ calls) by a different modem. In the rest of the channels there is one call belonging to a different modem in each window (see Figure 2). This situation is created if the first mW calls belong to different modems and afterwards calls to the modems occupying the first channel arrive one after the other until the channel is full. At this point we let each modem in the first channel receive $U - \frac{U}{W}$ more calls. All these calls are blocked by the algorithm since there is no way to move the existing calls to the same jitter window in any other channel. The number of calls that WF accepts is $WF(L) = U + (m-1)W$ while an optimal algorithm can accept $OPT(L) = min\{mU, WU + (m-1)W\}$. We choose $U > W(W+1)$ and $m = W + 1$ which means that all the calls are accepted by an optimal algorithm. The performance ratio in this example is

$$R_{WF} = \frac{U + (m-1)W}{WU + (m-1)W} = \frac{1}{W} + \frac{W-1}{U+W} \tag{7}$$

As $U \to \infty$ the performance ratio converges to $R_{WF} = \frac{1}{W}$.

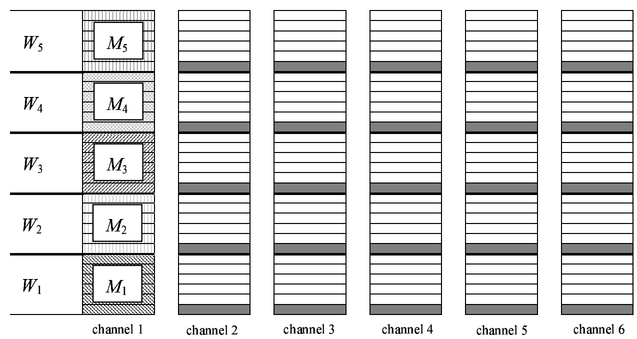

Fig. 2. Worst case example of algorithm WF for $U = 30$ and $W = 5$. Each modem $M_1, ..., M_5$ now receive 14 more calls.

Worst case example for Best-Fit and First-Fit

We create the situation depicted in Figure 3; the first channel is full and its first jitter window contains $\frac{U}{W}$ calls belonging to different modems. In the rest of the channels the first jitter window is full containing $\frac{U}{W}$ calls of the same modem (the construction of this situation by the algorithms is presented in [14]). At this point we let each modem in the first jitter window of the first channel receive $U - 1$ more calls. All these calls are blocked by the algorithm since there is no way to move the existing calls to the same jitter window in any other channel. The number of calls the algorithms accept is $BF(L) = FF(L) = U + (m-1)\frac{U}{W}$ while OPT can accept $(U-1)\frac{U}{W}$ additional calls, i.e., $OPT(L) = U + (m-1)\frac{U}{W} + (U-1)\frac{U}{W}$. We let the number of channels be $m = \frac{U}{W-1} + 1$ (we choose U for which m is an integer). The performance ratio in this case is

$$\begin{aligned} R_{BF} = R_{FF} &= \frac{U + (m-1)\frac{U}{W}}{U + (m-1)\frac{U}{W} + (U-1)\frac{U}{W}} \\ &< \frac{1}{W}\left(\frac{U+W}{U}\right) \end{aligned} \tag{8}$$

As $U \to \infty$ the performance ratio converges to $\frac{1}{W}$. ■

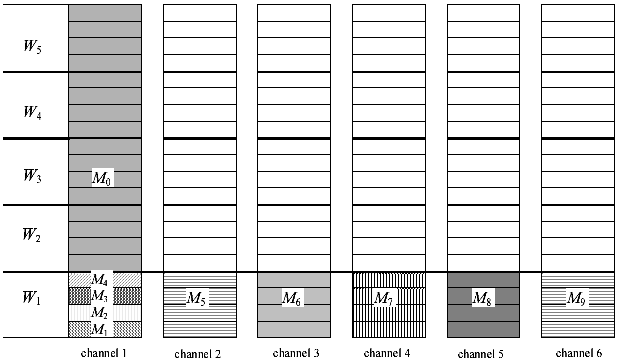

Fig. 3. Worst case example of algorithms BF and FF for $U = 20$ and $W = 5$. Each modem $M_1, ..., M_4$ now receive 19 more calls.

B. Scheduling without Using Jitter Windows

The use of jitter windows simplifies the scheduling algorithms. However, jitter windows add restrictions which are not imposed by the scheduling rules and may therefore degrade the performance of an algorithm. In this section we analyze the performance of our algorithms when jitter windows are not used. To provide a comparison with the case where jitter windows are used we define $W = I/J$. We show that as U increases the worst case performance ratio of the algorithms converges to $\frac{1}{W}$.

The example we present is suitable for all three algorithms. We create a situation in which the first channel is full containing k calls of different modems ($M_1, ..., M_k$) and $U - k$ calls of modem M_{k+1}. In each of the other channels there are $\frac{U}{W} + k$ calls of a single modem. It is easy to verify that this initial situation is possible in all three algorithms. In the next stage $U - k$ calls arrive to each modem in the first channel, i.e., $M_1, ..., M_{k+1}$. Since there is no way to switch a modem from channel 1 without violating its jitter constraints all these calls are blocked. The algorithm accepts $A(L) = U + (m-1)(\frac{U}{W} + k)$ calls while an optimal algorithm can accept $OPT(L) = \min\{(k+1)U + (m-1)(\frac{U}{W} + k),\ mU\}$. If we choose $k = \sqrt{U}$ (assuming it is an integer) and $m = \sqrt{U} + 1$ the performance ratio is

$$R_A = \frac{U + \sqrt{U}\left(\frac{U}{W} + \sqrt{U}\right)}{(\sqrt{U}+1)U} = \frac{\sqrt{U} + 2W}{W(\sqrt{U}+1)} \quad (9)$$

As U increases the performance ratio converges to $\frac{1}{W}$.

VII. Conclusions and Further Research

In this paper we considered the problem of scheduling packet telephony calls in a multi-channel cable network. We showed that the scheduling problem is NP-hard even in the case where all the calls have the same characteristics. We then suggested three simple approximation algorithms and investigated their performance. In the case where jitter constraints are not present our algorithms perform well. Their performance ratio is $2/3$ which we showed to be the best any deterministic algorithm can achieve. When jitter constraints are present, we showed that the performance ratio of the algorithms behaves as J/I, i.e., the ratio between the frame length and the tolerated jitter. For VoIP applications I is typically 10 ms while J is typically between 2 ms and 10 ms; these values give us a reasonable worst case performance ratio.

To evaluate the benefits of channel switching we showed that when channel switching is disabled the performance of the algorithms degrades severely. Supporting channel switching adds complexity to both the modem and the scheduler, our results indicate that the added complexity is worthwhile. With (considerable) additional complexity one can design a modem that would be able to have simultaneous calls on more than one channel. Our results can be used to evaluate the benefits of implementing such improvements.

The paper leaves plenty of ground for future research. Several extensions to this paper are found in [14] where we explore the scheduling problem under the (realistic) assumption that the number of simultaneous calls a modem may have is bounded. We also investigate the scheduling problem with temporary calls. There are many other subjects that await further research; extending the analysis for the case of non uniform calls, i.e., calls with different grant intervals and/or sizes is particularly challenging.

References

[1] Cable Television Laboratories Inc. CableLabs, *Data-Over-Cable Service Interface Specifications - Radio Frequency Interface Specification (status: interim)*, April 2000.

[2] "Cable television laboratories inc. CableLabs." `http://www.cablelabs.com`.

[3] J. Albrycht, "VoIP: putting the pieces together," *CED*, vol. 24, pp. 44–48, December 1998.

[4] B. Beser, "Providing enhanced services over DOCSIS," *CED*, vol. 27, pp. 80–84, April 2001.

[5] Cable Television Laboratories Inc. CableLabs, *PacketCable™ Dynamic Quality-of-Service Specification (status: interim)*, August 2000.

[6] D. Bushmitch, S. Mukherjee, S. Narayanan, M. Ratty, and S. Q. Beser, "Supporting MPEG video transport on DOCSIS-compliant cable networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 1581–1596, September 2000.

[7] D. Jianghong, X. Zhongyang, C. Hao, and D. Hui, "Scheduling algorithm for MPEG-2 TS multiplexers in CATV networks," *IEEE Transactions on Broadcasting*, vol. 46, pp. 249–255, December 2000.

[8] V. Sdralia, C. Smythe, P. Tzerefos, and S. Cvetkovic, "Performance characterization of the MCNS DOCSIS 1.0 CATV protocol with prioritized first come first served scheduling," *IEEE Transactions on Broadcasting*, vol. 45, pp. 196–205, June 1999.

[9] P. Tzerefos, V. Sdralia, C. Smythe, and S. Cvetkovic, "Delivery of low bit rate isochronous streams over the DOCSIS 1.0 cable television protocol," *IEEE Transactions on Broadcasting*, vol. 45, pp. 206–214, June 1999.

[10] N. Naaman and R. Rom, "Scheduling constant bit rate flows in data over cable networks," in *Proceeding of the 7th IEEE Symposium on Computers and Communications ISCC'02*, (Taormina, Italy), pp. 577–582, 2002.

[11] W. S. Lai, "Upstream bandwidth allocation for packet telephony in hybrid fiber-coax systems," in *Proceedings of the 2000 Symposium on Performance Evaluation of Computer and Telecommunication Systems*, (San Diego, CA, USA), pp. 96–100, 2000.

[12] A. Tripathi and K. N. Sivarajan, "Resource allocation in gprs wireless networks," in *Proceeding of the IEEE International Conference on Personal Wireless Communications*, (Piscataway, USA), pp. 388–394, 2000.

[13] ETSI Specifications, *Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Overall description of the GPRS radio interface; Stage 2 (GSM 03.64 version 7.0.0 release 1999)*.

[14] N. Naaman and R. Rom, "Scheduling constant bit rate flows over a TDMA channel," tech. rep., Technion EE, to appear.

[15] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: W H Freeman and Co., 1979.

[16] E. G. Coffman Jr., M. R. Garey, and D. S. Johnson, "Approximation algorithms for bin packing: A survey," in *Approximation Algorithms for NP-Hard Problems* (D. Hochbaum, ed.), pp. 46–93, Boston: PSW, 1996.