A Generic Quantitative Approach to the Scheduling of Synchronous Packets in a Shared Uplink Wireless Channel

Reuven Cohen Liran Katzir
Department of Computer Science
Technion
Israel

***Abstract*— The scheduling logic at the base station of a shared wireless medium supports time-dependent (synchronous) applications by allocating timely transmission grants. To this end it must take into account not only the deadlines of the pending packets, but also the channel conditions for each potential sender, the requirements of non-synchronous applications, and the packet retransmission strategy. With respect to these factors, we identify three scheduling scenarios and show that the scheduler logic faces a different challenge in addressing each of them. We then present a generic scheduling algorithm that translates all the factors relevant to each scenario into a common profit parameter, and selects the most profitable transmission instances.**

I. Introduction

A synchronous application like streaming (one-way voice/video) or telephony (two-way voice) is one that demands from the network guaranteed bandwidth, guaranteed maximum delay, and guaranteed loss rate. In wireless access networks there is a common channel that needs to be shared by many stations using a MAC (Medium Access Control) protocol. In this paper we propose a MAC layer uplink scheduling for synchronous traffic in wireless networks where the transmission of such traffic is governed by explicit grants allocated by the base station. Examples for such networks are IEEE 802.16 [8], [6] or IEEE 802.11 (in infrastructure mode). In the proposed scheme the base station allocates transmission grants to the end host of each synchronous call while taking into account the following scheduling considerations (SCs):

- SC1. The specific QoS requirements of each call: the grants should meet the negotiated grant size, grant periodicity, and tolerated grant jitter.
- SC2. The specific conditions of each uplink channel: basically, if a channel experiences bad SNR, the scheduler will try to delay the grant as much as possible.
- SC3. The specific Application layer loss recovery mechanism employed by each synchronous call codec. Several researchers have shown that some synchronous packets are more sensitive to loss than others [14], [15]. The quality of a synchronous call can therefore be improved by assigning a higher drop priority to the more important packets. For example, when media-dependent FEC is employed and a packet is lost due to a bad channel, the scheduler should increase the priority of the next packet from the same synchronous call, in order to increase the probability that this packet will be received on time.
- SC4. The specific MAC layer loss recovery mechanisms employed by the network, and in particular, whether ARQ is employed.
- SC5. Adaptive Modulation and Coding (AMC), along with power control.

We describe the proposed scheme in the context of a single upstream channel. In terms of the 802.16 standard [8], such a channel is provided by the single carrier PHY and by the OFDM PHY. We also extend the scheme to address multiple simultaneous transmissions on different sub-channels when OFDMA (or OFDM with sub-channelization) PHY is used.

The main idea behind the proposed scheme is to assign a profit to the transmission of each synchronous packet at every time slot, while taking into account all the relevant scheduling aspects. For example, the profit is proportional to the priority of the packet, to the distance to the packet due date, and to the probability of a successful transmission.

In the proposed scheme the scheduler has three tasks, as described in the following and summarized in Figure 1. The first (Scenario A in Figure 1) is to determine which of the synchronous packets should be dropped and which transmitted. This decision is important only if the channel bandwidth and the tolerated jitter cannot accommodate the demand of the synchronous applications. This is shown clearly in Figure 2, which depicts the loss rate of VoIP packets as a function of the normalized jitter, i.e., $\frac{\text{tolerated grant jitter}}{\text{packet transmission time}}$ for various loads. These graphs were obtained through simulations, assuming an error-free channel, 1-slot packets, and EDF (Earliest Deadline First) scheduling, which is known to be optimal for minimizing the number of packets that miss their deadlines under these conditions. When the normalized jitter is higher than 10, losses due to scheduling conflicts are not likely even if the load of the synchronous traffic is very close to 100%. Note that non-synchronous (best-effort) traffic has no effect on the graph because it is accommodated only when there is no synchronous traffic.

Scenario	Synch. load	Tolerated jitter	Scheduler challenge	Benefit gained from efficient scheduling
A	high	short "normalized jitter"	selecting the most important packets for transmission	on-time transmission of the most important synchronous packets
B	irrelevant	longer than error burst length	selecting the best time and PHY profile for each packet	successful transmission of more synchronous packets using less bandwidth
C	irrelevant	shorter than error burst length	minimizing the number of bad synchronous transmissions	(a) more available bandwidth for non-synchronous applications (b) successful transmission of more synchronous packets using less bandwidth

Fig. 1. The three tasks of the proposed scheduling algorithm

In a bad channel, the scheduler has an important task even if the load imposed by the synchronous traffic is low compared to the channel bandwidth. If the tolerated jitter is long enough compared to the average length of an error burst (Scenario B in Figure 1), as might be the case for video streaming, the second task of the scheduler is to determine the best combination of transmission time and AMC (Adaptive Modulation and Control) for each packet, in order to maximize the number of synchronous packets that are received on time, with no error using minimal bandwidth. When the tolerated jitter is not long enough (Scenario C in Figure 1), as in the case of packetized telephony, the scheduler does not have enough flexibility to wait until an error burst is likely to end. In that case, the third task of the scheduler is to determine which packets should be ignored, in order to minimize bandwidth waste. This can increase the available bandwidth for non-synchronous (best-effort) applications that experience a good channel.

The proposed quantitative-based approach is said to be generic because it is applicable for all the scenarios described in Figure 1, and for any combination of scheduling considerations SC1-SC5 discussed earlier.

The rest of this paper is organized as follows. In Section II we discuss related work. In Section III we present the proposed scheme. In Section IV we show how to compute the profit of transmitting a packet in every slot, with or without MAC layer retransmission support. In Section V we address some practical considerations related to the proposed scheme. Section VI presents a simulation study of the proposed scheme, and Section VII concludes the paper.

II. Related Work

We are not aware of previous works that address the scheduling problem while taking into account the aforementioned scheduling considerations. Ref. [1] proposes interesting scheduling algorithms that do take into consideration packet loss due to transmission errors. It also provides a general guideline for addressing SC2 and SC4 using the notion of "backoff time." The idea is that if a call has experienced a recent loss due to transmission errors, then a new packet generated by this call at time t_0 and having a deadline of t_1 will be scheduled during $[(t_0 + t_1)/2, t_1]$ rather than during $[t_0, t_1]$, in order to allow the channel to recover. However,

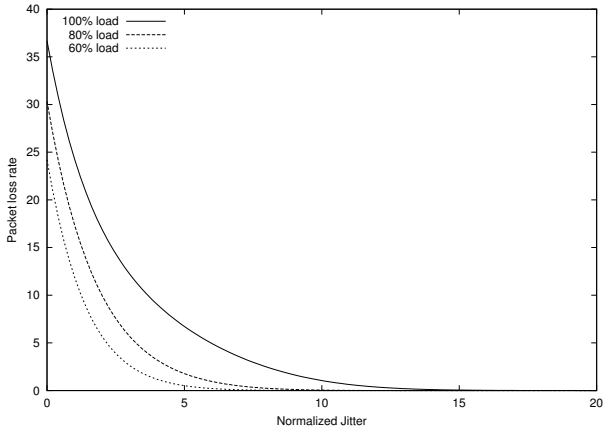


Fig. 2. The loss rate vs. jitter and load for VoIP packets

this scheme is only mentioned as a possible strategy and its performance is not discussed.

In [12], a model for wireless fair scheduling based on the adaptation of fluid fair queuing is proposed. This model is different from the one proposed in this paper. While we assume that the base station knows exactly when each host has a new packet awaiting transmission, this assumption is not made in [12]. Moreover, [12] does not take into account issues related to SC3-SC5.

In [16] it is shown that when EDF (Earliest Deadline First) is implemented over channels that are in good condition, the number of packets lost due to deadline expiration is minimized. This result has a bearing on congested channels, for which it is important to determine which of the packets should be transmitted and which should be dropped. This is in contrast to the problem considered in this paper where we determine the optimal time for transmitting each packet under not necessarily congested conditions. In [2], a scheduling algorithm that uses an N-state Markov model, where $N > 2$, to characterize the channel is presented. This algorithm supports AMC in order to adjust the modulation and FEC to the forecasted channel state.

A common way to address SC2 and SC5 is by assigning higher data rates to hosts with a better channel, in order to maximize throughput while ensuring acceptable bit-error rate

(BER) [7], [13]. In the uplink channel of an OFDM network, multiple hosts can transmit simultaneously over different sub-carriers. Since the channel characteristics for different users may be independent, dynamic assignment of sub-carriers to hosts can significantly improve the throughput [19], [18]. However, this "water filling" approach for maximizing instantaneous throughput does not take into account the QoS requirements of the calls originating these packets, and it is therefore unsuitable for synchronous traffic. The authors of [10] address this problem, in the context of OFDMA, by asking how, given a set of hosts and a set of sub-carriers, the sub-carriers should be allocated to the hosts in a way that satisfies the rate requirements of each host while using minimum power. In [17] utility-based cross-layer optimization problems are defined using the channel model, utility functions, adaptive modulation and frequency power allocation.

III. The Quantitative-Based Framework

The scheme proposed in this paper is based upon quantitative rather than qualitative considerations. For each "scheduling interval," namely, an interval of time for which scheduling decisions are made by the base station for all the packets available for transmission, the scheduler determines the expected profit for scheduling each packet in every time slot. Then, the schedule with maximum expected profit is chosen. The most difficult part of this scheme is finding a good profit function that takes into account the various considerations.

To simplify the presentation of our quantitative framework, we present it incrementally. We start with the basic model, referred to as Model 1. In this model the channel is assumed to be clean, and only the QoS requirements of each call (SC1) and the profit of each packet in the context of its specific call (SC3) are taken into consideration. We then consider Model 2, where the channel conditions for each host (SC2) are also taken into account. Model 3 extends the framework to accommodate multiple PHY profiles (SC5), and Model 4 addresses MAC layer retransmissions (SC4) as well.

Model 1: Assume that the head-end executes the scheduling algorithm every scheduling interval of T uplink slots. T is usually equal to the length of the uplink frame (a few milliseconds). However, its value can be dynamically adjusted by the base station. For example, in order to reduce the load imposed on the base station, this value can be extended to several frame times. The penalty in such a case is slower response to the first packet of a new flow or to the first packet of a new talkspurt when silence suppression is used. The base station maintains a profit matrix ϕ. Entry $\phi[c,t]$ in this matrix indicates the profit if the first pending packet of call c is transmitted starting from slot t and is correctly received by the base station. If the transmission starting at slot t cannot be completed before the deadline of the packet, the profit is 0; otherwise, the profit is 1. In this case, by finding a schedule that maximizes the profit, we maximize the number of packets that meet their QoS requirement, thereby addressing SC1. However, in order to take into consideration not only SC1 but also SC3, ϕ can be viewed as a non-binary profit function that takes into account the priority of the packet, as determined by upper layer information. This priority can be modified by the scheduler dynamically. For instance, it can be increased if a previous packet of the same call was not received on time.

It will be convenient to assume in the meantime that each synchronous call has only one pending packet during each scheduling interval. A pending packet is a packet that (1) was released, (2) has not yet been successfully transmitted, and (3) whose due date has not yet expired. This assumption holds only when the tolerated jitter for a call is shorter than the packetization time (i.e., packet inter-arrival time).

Model 2: We assume that at slot t there is a probability $\lambda(c,t) \geq 0$ that the transmission from the host of call c over the shared channel will *not* be lost due to a transmission error. The optimization criterion is to maximize the sum of the profits of packets that are transmitted on time and experience no transmission error.

Suppose that the base station is able to compute the value of $\lambda(c,t)$ for every time slot t and for every synchronous call c. The profit matrix ϕ used for the previous model is replaced by a new expected profit matrix μ, where $\mu[c,t] = \phi[c,t] * \lambda(c,t)$. We later show how $\lambda(c,t)$ can be computed for fixed and for mobile hosts.

A schedule σ is a transmission vector that indicates which packet should start being transmitted in which slot. If $\sigma(t) = c$, then at time slot t the transmission of the current packet of call c should start. The overall profit gained from a schedule σ is $\text{Profit}(\sigma) = \sum_{t=1}^{T} \mu[\sigma(t), t]$, where $[1 \cdots T]$ is the *scheduling interval.* We seek a schedule σ for which $\text{Profit}(\sigma)$ is maximum. Algorithms for finding the best schedule for a given profit matrix μ are discussed in Section V.

There is one problem with employing the profit-based framework in Model 2. Suppose that the channel of call c is in bad condition when a packet is released. If the shared channel is not heavily loaded, an algorithm that seeks to maximize the expected profit will choose to transmit the packet at a slot where the error probability is the smallest, even if this probability is still very high, just because there are no other waiting synchronous packets. However, a better decision is not to schedule this packet during the current scheduling interval, but to wait for one of the subsequent scheduling intervals, where the probability for an error might be smaller. The vacant slots can then be used for the transmission of some best-effort packets. There are several possible approaches for addressing this problem:

1) To change the optimization criterion from maximizing the aggregated expected profit to maximizing the average expected profit per slot. This criterion penalizes the scheduler for transmitting packets in bad slots. The drawback of this approach is that the scheduler will avoid transmitting a packet in a bad channel even if this packet is very close to its deadline.
2) To determine a minimum threshold Δ for the probability of a successful transmission. When this probability is smaller than Δ, the expected profit is set to 0, and the scheduler will not select the packet for transmission. The

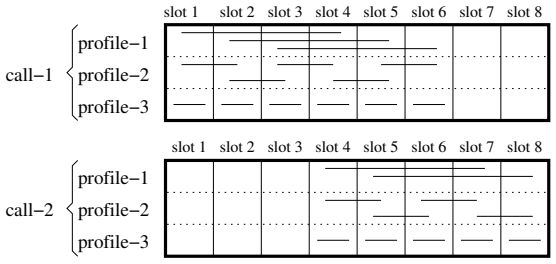


Fig. 3. A 3D (2x3x8) expected profit matrix μ with 2 calls, 3 PHY profiles and 8 slots

value of Δ can be dynamically adjusted according to the load of the non-synchronous traffic.

3) To run the algorithm for a period of T' slots, longer than the scheduling interval T, while implementing only its short-term decisions. Namely, we find the best packet to be transmitted during each of the next T' slots, but take into consideration only the decisions made for the first T slots, while ignoring those made for slots $[T+1 \cdots T']$. The next time the scheduler is invoked, after a time equivalent to T slots, the matrix will contain all the packets not transmitted before, including those scheduled to be transmitted during slots $[T+1 \cdots T']$. The fraction T'/T is considered as the "lookahead ratio." The main drawback of this approach is that it increases the running time of the scheduler by a factor of T'/T (see Section V).

Model 3: The various PHY layer FEC options are paired with modulation schemes, like QPSK, 16-QAM and 64-QAM, to form a pre-defined set of PHY profiles with varying robustness and efficiency.

To accommodate this model, the expected profit matrix μ is extended to 3 dimensions: calls, PHY profiles, and slots. Entry $\mu[c,t,m]$ in this matrix is set to $\phi[c,t] * \lambda(c,t,m)$, where $\lambda(c,t,m)$ is the probability that the packet of call c will be transmitted correctly starting from slot t using PHY profile m. Figure 3 shows the expected profit matrix μ with 2 calls and 3 PHY profiles. The packet of call-1 is available for transmission in slot 1. It must be transmitted not later than slot 6 in order to meet its tolerated jitter. This packet requires a bandwidth of 4 slots for the most robust PHY profile (profile-1), 2 slots for the less robust profile, and only one slot for the least robust profile. As the figure shows, there are 3 possible *transmission instances* for this packet using profile-1, 5 possible transmission instances using profile-2, and 6 possible transmission instances using profile-3. From these 14 possible transmission instances, the scheduler is allowed to select at most one, and it must not select any other packet for transmission during the same slot(s). The expected profit of each instance is not shown in the figure. It depends both on the exact time schedule for the instance, as discussed later, and on the PHY profile of the instance. The packet of call-2 is of a similar size and tolerated jitter. However, it is available only in slot 4, and it therefore has only 2 profile-1 transmission instances, 4 profile-2 transmission instances, and 5 profile-3 transmission instances.

Model 4: This is the most generic model, which also takes into account MAC layer retransmissions (ARQ), when supported. In this model, we add to the probability of a successful transmission at t the conditional probability of successful retransmission(s). □

In Model 4, the expected profit when the number of retransmissions is only bounded by the tolerated jitter can be expressed as:

$$\phi[c,*] \cdot \sum_{i=1}^{R} [\text{Prob}(i\text{'th transmission is good} \mid \text{previous } i-1 \text{ transmissions are faulty}) \cdot \text{Prob}(i-1 \text{ consecutive faulty transmissions})], \quad (1)$$

where R is the maximum number of transmissions. In the next section we will elaborate on this simplified equation.

IV. Computing the Probability for Successful Transmissions and Retransmissions

A. Transmissions by Stationary and Mobile Hosts

For static hosts, the process of packet loss in a wireless channel can be modeled with a good approximation by a low order Markovian chain, such as a two state Gilbert model [11], [20]: one state, referred to as state '1', represents a bad channel, while the other state, referred to as '0', represents a good channel. Let $S(n) \in \{0,1\}$ be the state during slot n. Let $Prob\,[S(n+1)=0 \mid S(n)=0] = p$ and $Prob\,[S(n+1)=1 \mid S(n)=1] = q$. The following discussion pertains to each specific host H. Let time 0 be the last time when H transmits any packet, not necessarily of a synchronous call, to the base station. The base station knows whether this transmission was good or bad, and it needs to compute the probability that the channel is in a bad state at time n as a function of the channel condition at time 0.

Let $T(n)$ be the probability that the channel is in error state at time n (i.e., $T(n) = Prob\,[S(n)=1]$). Hence, we have

$$T(n+1) = qT(n) + (1-p)(1-T(n))$$

or equivalently

$$T(n+1) = (q+p-1)T(n) + (1-p).$$

The solution for $T(n+1) = aT(n) + b$, where $T(0) = C$ is

$$T(n) = Ca^n + ba^{n-1} + ba^{n-2} + \cdots + b = Ca^n + b\sum_{i=0}^{n-1} a^i. \quad (2)$$

Assuming that $a \neq 1$, for $n > 0$ we get

$$T(n) = Ca^n + b\frac{a^n - 1}{a-1}$$

and therefore

$$T(n) = C(p+q-1)^n + (1-p)\frac{(p+q-1)^n - 1}{p+q-2}. \quad (3)$$

To find $Prob\,[S(n) = 1 \mid S(0) = 1]$, we substitute $C = 1$ into Eq.(3) and get

$$Prob\,[S(n) = 1 \mid S(0) = 1] = \\ = (p+q-1)^n + (1-p)\frac{(p+q-1)^n - 1}{p+q-2}. \quad (4)$$

To find $Prob\,[S(n) = 1 \mid S(0) = 0]$ we substitute $C = 0$ into Eq.(3) and get

$$Prob\,[S(n) = 1 \mid S(0) = 0] = \\ = (1-p)\frac{(p+q-1)^n - 1}{p+q-2}. \quad (5)$$

For static hosts, the values of p, q, and the probability for a bit error in the bad (1) state, referred to as *Err*, can be computed using statistical information from each channel. However, this model is not valid for mobile hosts, because the quality of the channel for such hosts is unstable. We now propose a heuristic that allows the base station to approximate the value of $\lambda(c, t, m)$, using the outcome of the latest uplink transmission of each call, without any knowledge of p and q for the associated host. Rather, the base station takes into account the status of the channel when a packet was last transmitted by the host.

Suppose that the last packet sent by the host on the uplink channel encountered a transmission error. This packet does not necessarily belong to the same synchronous call. It might be a best-effort packet belonging to another application at the considered host, a synchronous packet of another call originating at the same host, or a special control message sent by each host periodically (like the *ranging* messages used by 802.16 [8]). Suppose that retransmission is not supported, and we therefore allow the host to transmit each synchronous packet only once. From Eq. 4 it follows that

$$Prob\,[S(n) = 0 \mid S(0) = 1] = \\ \frac{(p+q-1)^n(1-q) + q - 1}{p+q-2}. \quad (6)$$

Assuming that $p + q \geq 1$, this probability increases with the value of n, implying that the maximum is achieved if the packet is scheduled as close as possible to its deadline. However, if the previous uplink transmission of the considered host was successful, then from Eq. 5 it follows that the packet should be transmitted as close as possible to its release time.

B. Retransmission Scheduling

Now, suppose that ARQ is supported. Consider first the case where only one retransmission is possible (e.g., due to relatively short tolerated jitter). Let Release(P) be the time the packet is ready for transmission and Deadline(P) be the last time at which the packet can be transmitted while still relevant at the receiving side. Suppose that a decision regarding the scheduling of the first transmission of packet P has to be taken at time t. Let t_0 be the last time before t when another packet is transmitted by the considered host, successfully or not. Without loss of generality, let Deadline$(P) - t_0 = N + 1$ slots. Let i and j be the time when the first and second transmissions should take place, respectively. The two possible relationships between t, t_0, i, j, Release(P) and Deadline(P), are shown in Figure 4.

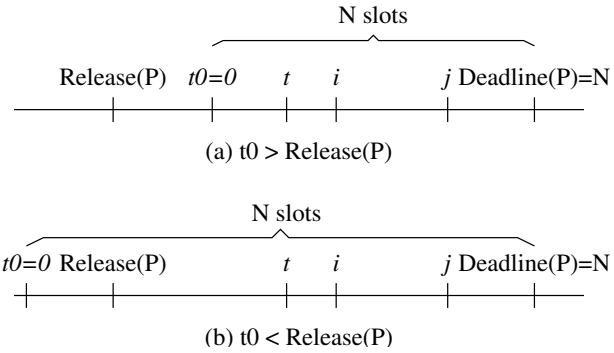


Fig. 4. Possible relationships between t_0 and Release(P)

If we substitute $R = 1$ into Eq. 1, the probability for a successful transmission is equal to the probability that the first transmission is good plus the conditional probability that the second transmission is good if the first transmission encounters an error. Let this sum be represented by $\mathcal{F}_g(i,j)$ if the channel is known to be good at t_0, and by $\mathcal{F}_b(i,j)$ if the channel is known to be bad at t_0. Hence, we have

$$\mathcal{F}_g(i,j) = Prob\,[S(i) = 0 \mid S(0) = 0] + \\ + Prob\,[S(i) = 1 \mid S(0) = 0] \cdot \\ \cdot Prob\,[S(j) = 0 \mid S(i) = 1], \quad (7)$$

and

$$\mathcal{F}_b(i,j) = Prob\,[S(i) = 0 \mid S(0) = 1] + \\ + Prob\,[S(i) = 1 \mid S(0) = 1] \cdot \\ Prob\,[S(j) = 0 \mid S(i) = 1] \quad (8)$$

We now want to determine the values of i and j that maximize $\mathcal{F}_g(i,j)$ and the values of i and j that maximize $\mathcal{F}_b(i,j)$. By substituting Eq. 4 and Eq. 5 into Eq. 7 we find that $\mathcal{F}_g(i,j)$ is maximized when i =*max*(Release$(P), t_0$) and $j =$ Deadline(P). By substituting Eq. 4 and Eq. 5 into Eq. 8 we find that $\mathcal{F}_b(i,j)$ is maximized when $j =$ Deadline(P). By substituting $j = N$ into Eq. 8, differentiating $\mathcal{F}_b(i,N)$ with respect to i, and equating to 0, we find that

$$\mathrm{MAX}_{i=\mathrm{Release}(P)}^{\mathrm{Deadline}(P)}(\mathcal{F}_b(i, j = N)) = \\ = \begin{cases} N/2 & \text{if } N/2 > \mathrm{Release}(P) \\ \mathrm{Release}(P) & \text{else.} \end{cases} \quad (9)$$

The case where $N/2$ >Release(P) and the case where $N/2 \leq$Release(P) are shown in Figure 5. Note that the case where $t_0 >$ Release(P) is equivalent to case (a).

We now extend this result to an arbitrary number of possible retransmissions.

Theorem 1: Suppose that at time t a scheduling decision has to be made for packet P. Suppose that the last transmission of any packet by the same host took place at $t_0 < t$. Then,

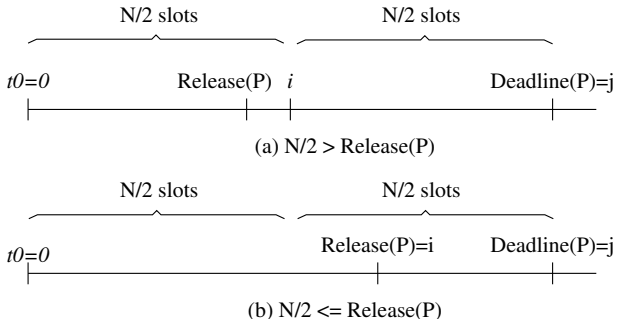


Fig. 5. The two cases of Eq. 9

(1) If the last transmission (at t_0) was unsuccessful, the probability of a successful transmission of P should be maximized as follows: (a) if only 1 additional transmission of P is allowed, this transmission should be scheduled as close as possible to Deadline(P); (b) if $X \geq 2$ additional transmissions of P are allowed, the next one should take place at $\max\{\text{Release}(P), t_0 + \frac{\text{Deadline}(P)-t_0}{X}\}$.

(2) If the last transmission (at t_0) was successful, then, regardless of the number of allowed retransmissions, the probability of a successful transmission of P should be maximized by scheduling it as close as possible to $\max\{\text{Release}(P), t_0\}$.

Proof: Let $\mathcal{G}_b(X,N)$ be the probability for a successful transmission of P at $t \geq \text{Release}(P)$, assuming that (a) the last transmission before t, at t_0, was bad, (b) X additional transmissions of P are allowed, and (c) Deadline(P) $- t_0 = N$. Let $\mathcal{G}_g(X,N)$ be a similar function, except that the last transmission (at t_0) was good.

Part 1(a) of the theorem follows directly from Eq. 6. We prove part 1(b) by induction on X. The induction basis is for $X = 2$, and it follows from Eq. 9. Assuming that the claim holds for $X-1$ transmissions, we now prove it for X transmissions. Without loss of generality, let $t_0+\alpha$ be the time when the first transmission takes place. Then,

$$\begin{aligned}\mathcal{G}_b(X,N) = Prob\left[S(\alpha)=0 \mid S(0)=1\right] + \\ + Prob\left[S(\alpha)=1 \mid S(0)=1\right] \cdot \\ \cdot \mathcal{G}_b(X-1, N-\alpha)\end{aligned} \tag{10}$$

For a given value of α, this equation is maximized when $\mathcal{G}_b(X-1,N-\alpha)$ is maximized. By the induction assumption, the maximum of $\mathcal{G}_b(X-1,N-\alpha)$ is achieved when the first transmission takes place at $t_0+\alpha+(N-\alpha)/(X-1)$, because $t_0+\alpha$ is the time when the last transmission took place. Note that Release(P) must be earlier than t_0 and hence $\max\{\text{Release}(P), t_0+\alpha+(N-\alpha)/(X-1)\}$ is $t_0+\alpha+(N-\alpha)/(X-1)$. Therefore, we have

$$\begin{aligned}\mathcal{G}_b^{max}(X-1,N-\alpha) = \\ Prob\left[S\left(\tfrac{N-\alpha}{X-1}\right)=0 \mid S(0)=1\right] + \\ + Prob\left[S\left(\tfrac{N-\alpha}{X-1}\right)=1 \mid S(0)=1\right] \cdot \\ \cdot \mathcal{G}_b\left(X-2, \tfrac{(N-\alpha)(X-2)}{X-1}\right)\end{aligned}$$

Since both $Prob\left[S\left(\frac{N-\alpha}{X-1}\right)=0 \mid S(0)=1\right]$ and $Prob\left[S\left(\frac{N-\alpha}{X-1}\right)=1 \mid S(0)=1\right]$ are constant, it is clear that $\mathcal{G}_b(X-1,N-\alpha)$ gets its maximum when $\mathcal{G}_b\left(X-2, \frac{(N-\alpha)(X-2)}{X-1}\right)$ gets its maximum. We can now use the induction assumption once again to find when this happens. By repeating this process $X-1$ times, we get that $\mathcal{G}_b(X-1,N-\alpha)^{max}$ has the same form as found for $T(n)$ in Eq. 2, namely,

$$\mathcal{G}_b^{max}(X-1,N-\alpha) = Ca^{X-1} + b\frac{a^{X-1}-1}{a-1}, \tag{11}$$

where $a = Prob\left[S\left(\frac{N-\alpha}{X-1}\right)=1 \mid S(0)=1\right]$, $b = Prob\left[S\left(\frac{N-\alpha}{X-1}\right)=0 \mid S(0)=1\right]$, and $C = \mathcal{G}_b(0,*) = 0$. Substituting this equation into Eq. 10, differentiating it with respect to α, and then equating the result to 0 yield that if Release(P) $> t_0 + \frac{N}{X}$ then $\mathcal{G}_b(X,N)$ gets its maximum at $\alpha = \text{Release}(P) - t_0$, whereas if Release($P$) $\leq t_0 + \frac{N}{X}$ then $\mathcal{G}_b(X,N)$ gets its maximum at $\alpha = \frac{N}{X}$. This completes the proof of 1(b).

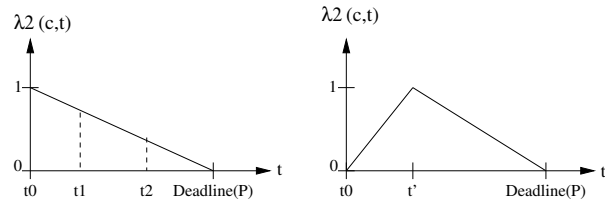
To prove part (2), note that the probability for success if X transmissions are allowed and the channel is known to be good at t_0 is given by

$$\begin{aligned}\mathcal{G}_g(X,N) = Prob\left[S(\alpha)=0 \mid S(0)=0\right] + \\ + Prob\left[S(\alpha)=1 \mid S(0)=0\right] \cdot \\ \cdot \mathcal{G}_b(X-1, N-\alpha)\end{aligned} \tag{12}$$

By substituting Eq. 11 into this equation, we find that the maximum is achieved for $\alpha = 0$. ■

As an example for using the results of Theorem 1, consider a packet P where Release(P) $= t_1$ and Deadline(P) $= t_2$. Suppose that a scheduling decision has to be made for packet P at time $t = t1$. Suppose that the last transmission of the same host before time t was at t_0, and that this transmission was unsuccessful. Assuming that up to N transmissions are allowed, the best time to schedule the first transmission of P is $\max\{t_1, t_3 = t_0 + \frac{t_2-t_0}{N}\}$. Suppose that $t_3 > t_1$ and consider the following sub-cases of this scenario:

1) Suppose that the same host transmits another packet at $t_4 \in [t_1, t_3]$ successfully. Then, the optimal time for transmitting P is shifted to t_4.
2) Suppose that the same host transmits another packet at $t_4 \in [t_1, t_3]$ unsuccessfully. Then, the optimal time for transmitting P is shifted to $\frac{t_2-t_4}{N}$.



(a) last transmission was good (b) last transmission was bad

Fig. 6. The function $\lambda_2(c,t)$

3) Suppose that the same host does not transmit another packet during $[t_0, t_3]$. Suppose also that due to scheduling conflicts P can only be scheduled for transmission at $t_4 > t_3$, and this transmission is unsuccessful. Then, the best time for scheduling the second transmission of P is at $t_4 + \frac{t_2 - t_4}{N-1}$.

Using these rules, we now show how the profit matrix μ is updated for every synchronous call c and every time slot of a scheduling interval $[t_1, t_2]$, assuming no knowledge regarding the values of p, q and *Err*. For each packet this algorithm only needs to know whether the last transmission on the same channel, not necessarily by the same synchronous call, was successful. The algorithm uses a continuous linear increasing or linear decreasing function $\lambda_2(t)$ whose value is 1 at the optimal slot and 0 at the sub-optimal slot.

Algorithm 1: filling up the profit matrix μ
Let $\phi[c,t]$ denote the "basic profit" of the next packet of call c, based on the content of the packet, its deadline, and the loss history of the call. For every $t \in [t_1,\ \min\{\text{Deadline}(P), t_2\}]$, set:

$$\mu[c,t,m] \leftarrow \phi[c,t] \cdot \lambda(c,t,m),$$

where $\lambda(c,t,m) = \lambda_1(c,m) \cdot \lambda_2(c,t)$. Function $\lambda_1(c,m)$ is the probability for success when transmitting in a good channel using modulation m, whereas function $\lambda_2(c,t)$ is determined as follows:

(a) If the last transmission by the same host was successful, then according to Theorem 1 the optimal transmission time is as early as possible. Hence, for every $t \in [t_1 \min\{\text{Deadline}(P), t_2\}]$ $\lambda_2(c,t) = \frac{\text{Deadline}(P)-t}{\text{Deadline}(P)-t_0}$. This function is depicted in Figure 6(a).

(b) If the last transmission by the same host, say at time t_0, was unsuccessful, and the packet can be transmitted at most $X \geq 1$ additional times, then according to Theorem 1, the optimal transmission time is $t' = \max(t_1,\ t_0 + \frac{\text{Deadline}(P)-t_0}{X})$. Hence, (i) for every $t \in [t_1, t']$, $\lambda_2(c,t) = \frac{t-t_0}{t'-t_0}$; (ii) for every $t \in [t', \min\{\text{Deadline}(P), t_2\}]$ $\lambda_2(c,t) = \frac{\text{Deadline}(P)-t}{\text{Deadline}(P)-t'}$. This function is depicted in Figure 6(b). □

V. Algorithms for Finding the Optimal Schedule in μ and Some Practical Considerations

A. Scheduling Algorithms

So far we have focused on how the matrix μ should be configured such that $\mu[c,t,m]$ will reflect the profit of transmitting the packet of synchronous call c at time t using PHY profile m. However, after μ is configured, the base station needs to run an algorithm for finding an optimal schedule in μ for the next T slots. The schedule indicates which packet should be transmitted during each time slot and using which PHY profile, and it can be viewed as a set of transmission instances.

In the hypothetical case where each synchronous packet fits a single slot and there is only one PHY profile, an optimal schedule can be found using the concept of maximum matching in a bipartite graph. However, in the case where packets are of arbitrary length, the problem of finding an optimal schedule is NP-complete (when only one PHY profile is used, the problem is equivalent to the problem discussed in [4]). In what follows we discuss two possible scheduling algorithms. The first one is a greedy algorithm. It scans the matrix μ and chooses a transmission instance with the maximum normalized profit (profit per slot). Recall that a transmission instance is a combination of a call, a PHY profile and several consecutive time slots. It then removes from matrix μ all the transmission instances that collide with the chosen one, i.e., all other instances of the same packet (call), and all the instances that use one or more of the slots used by the selected packet. This process is repeated until μ is empty. The time complexity of this algorithm using naive implementations is $O(T' \cdot C \cdot P \log(T' \cdot C \cdot P))$, where T', C and P are the dimensions of matrix μ: T'/T is the lookahead ratio where T is the number of slots in a scheduling interval, C is the number of active calls, and P is the number of PHY profiles.

While the greedy algorithm is easy to implement, there is no upper bound on its worst-case performance compared to the optimal solution (which cannot be found using a polynomial time algorithm since the problem is NP-Complete). As we show in [4], the following algorithm guarantees a solution whose profit in the worst case is not less than 1/2 of the maximum profit:

Algorithm 2: finding an optimal schedule in μ

1) set $i \leftarrow 1$.
2) Find in μ the transmission instance that ends first, and choose it to be I_i. If two or more transmission instances meet this requirement, select one of them arbitrarily.
3) Decrease in μ the profit of instance I_i and all the transmission instances that have a conflict with I_i (i.e., all the instances of the same packet with a different combination of PHY profile and transmission time, and all the instances of other packets whose transmission time overlaps the transmission time of I_i) by the profit of transmission instance I_i.
4) Remove from μ all the transmission instances whose profit is ≤ 0. That is, instance I_i as well as any instance

that has a conflict with I_i and whose profit before step 3 was executed was smaller than the profit of I_i.

5) If μ is not empty, set $i \leftarrow i+1$ and go to step 2, otherwise let $K = i$, and σ=NIL.
6) We now create from $\{I_1 \cdots I_K\}$ a feasible schedule σ in the following way: for $i = K$ to 1 do: if $\sigma \cup \{I_i\}$ is feasible (that is, I_i has no conflict with any instance in σ), then $\sigma \leftarrow \sigma \cup \{I_i\}$.

Careful implementation of this algorithm would result in running time complexity of $O(T' \cdot C \cdot P)$.

In OFDMA systems [9] multiple hosts can be scheduled to concurrently transmit on the upstream channel using different sub-channels. The PHY decoding process employed by the base station does not distinguish between the receiving time of packets transmitted in different time-slots and/or different sub-channels of the same uplink frame. Recall that the length of a scheduling interval is T slots, and that a lookahead ratio of T'/T, where $T' \geq T$, can be used. Suppose that each scheduling interval consists of an integer number $F \geq 1$ uplink frames, where the size of each frame is T/F slots. While the two algorithms discussed above determine which *individual* packet will be transmitted during each of the T *slots,* they should be modified in order to determine which *set* of packets will be transmitted during each of the F *frames.* Of course, one could also use the two algorithms discussed above for this problem by ignoring the exact slots assigned to each packet. However, this algorithm is no more polynomial in the size of the input ($(T' \cdot F/T) \cdot C \cdot P$). We can reduce the running time complexity of the scheduling algorithm by maintaining a matrix whose dimensions are $F \cdot T'/T$, C, P rather than T' , C, P, and by applying algorithms for the "Generalized Assignment Problem" (GAP) in order to determine the set of packets to be transmitted during each of the $T' \cdot F/T$ frames, rather than during each of the T' slots. A discussion on the GAP problem, which is also NP-complete, as well as possible approximation algorithms, are presented in [3].

B. Tolerated Jitter vs. Packetization Time

So far it has been assumed that each call has only one pending packet. As already noted, this assumption does not hold when the packetization time is smaller than the tolerated grant jitter. Figure 7(a) shows a case where the tolerated jitter is 2/3 of the packetization time and Figure 7(b) shows a case where the tolerated jitter is twice the packetization time. The former case is typical for conversational voice, while the latter is typical for one-way video sessions.

There are several approaches to accommodate multiple pending packets per call. In what follows we distinguish between the case where packet retransmission is not supported and the case where it is supported. In the former case, the matrix μ will contain a row for each pending packet of each call rather than a row for each call. It is therefore possible for the scheduler to choose two or even more packets from the same call during the same scheduling interval. Since retransmissions are not allowed, there is no problem with transmitting the ith packet of a call before the host knows

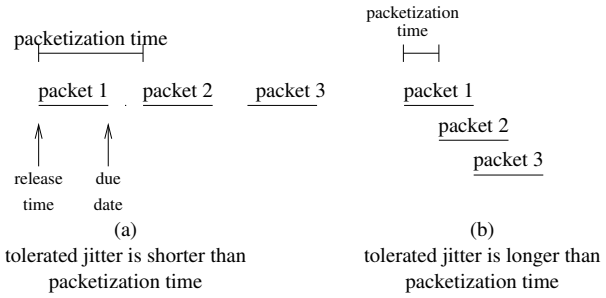
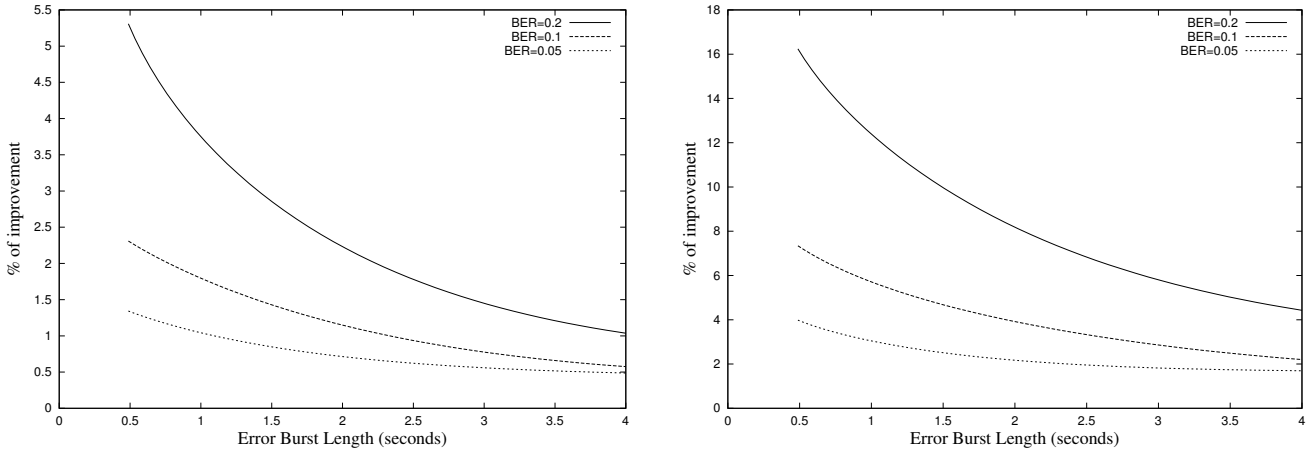


Fig. 7. Tolerated jitter vs. packetization time

whether the $(i-1)$th packet has been correctly received. However, we need to make sure that packets of the same call are not transmitted out of order. While it is likely that the scheduler will schedule packets from the same call in their original order, this condition is not guaranteed in the general case because an older packet has a closer due date and therefore a higher profit. This issue can be addressed by reordering the scheduler output. The advantage of having an entry in μ for each outstanding packet, rather than for each call, is that the scheduler is able to benefit from periods when the channel's condition has changed from bad to good by allocating to a single call multiple grants during the same scheduling interval.

However, when MAC layer retransmissions are allowed, this solution is no longer applicable. In order to guarantee that packets of the same call are received in their original order, the host should not transmit a packet before it knows that earlier packets from the same call have been correctly received. Hence, for this case matrix μ should contain again a single entry per call, and this entry should indicate the profit for transmitting the oldest pending packet of this call in each time slot. For calls with multiple pending packets and ARQ support, the retransmission interval for each packet must also be determined. If we allow the first pending packet to be retransmitted during the maximum possible interval, we reduce the period of time during which the next pending packets can be scheduled. However, the following observation explains why such a policy still works well. The difference between the due times of two successive packets is always $\geq$ packetization time. By the rules described in Algorithm 1, in the worst case the last transmission of the first pending packet might take place just before this packet's deadline. If this transmission is bad, the channel has been bad for a long period of time, so it does not matter which packets have been scheduled for transmission during this interval. On the other hand, if the transmission is good, the burst of pending packets awaiting transmission can be accommodated. Each of these packets has an "independent tolerated jitter," i.e., a tolerated jitter that is not affected by the scheduling of previous packets from the same call. This tolerated jitter is equal to the packetization time, and is much longer than the packet transmission time. Hence, by the graph in Figure 2, the scheduler is likely to schedule the whole burst on time with no problem.



(a) No retransmission (b) One retransmission

Fig. 8. The reduction in the bandwidth consumed by the quantitative algorithm vs. the reference algorithm when tolerated jitter is 1 sec.

VI. Simulation Results

In this section we present simulation results for some of the scenarios discussed in Section I. We start with Scenario B, where the tolerated jitter is long (or at least not too short) compared to the length of an error burst, and the load of the synchronous traffic is not necessarily high. Recall that the motivation for using a smart scheduler in this scenario is two-fold: to increase the number of packets successfully received by the hosts, and to decrease the bandwidth used for these packets.

As already said, we are not aware of any previous scheme that addresses scheduling considerations SC1-SC5 together. Hence, we compare the results of the proposed quantitative approach without full knowledge of the channel condition (that is, Algorithm 1), to the results of a reference algorithm that uses EDF policy in order to schedule the first copy of each packet. As proposed in [1], if the ith copy is lost at time t, the reference algorithm schedules the $(i+1)$th copy for transmission at $t + 0.5(\text{Deadline}(P) - t)$.

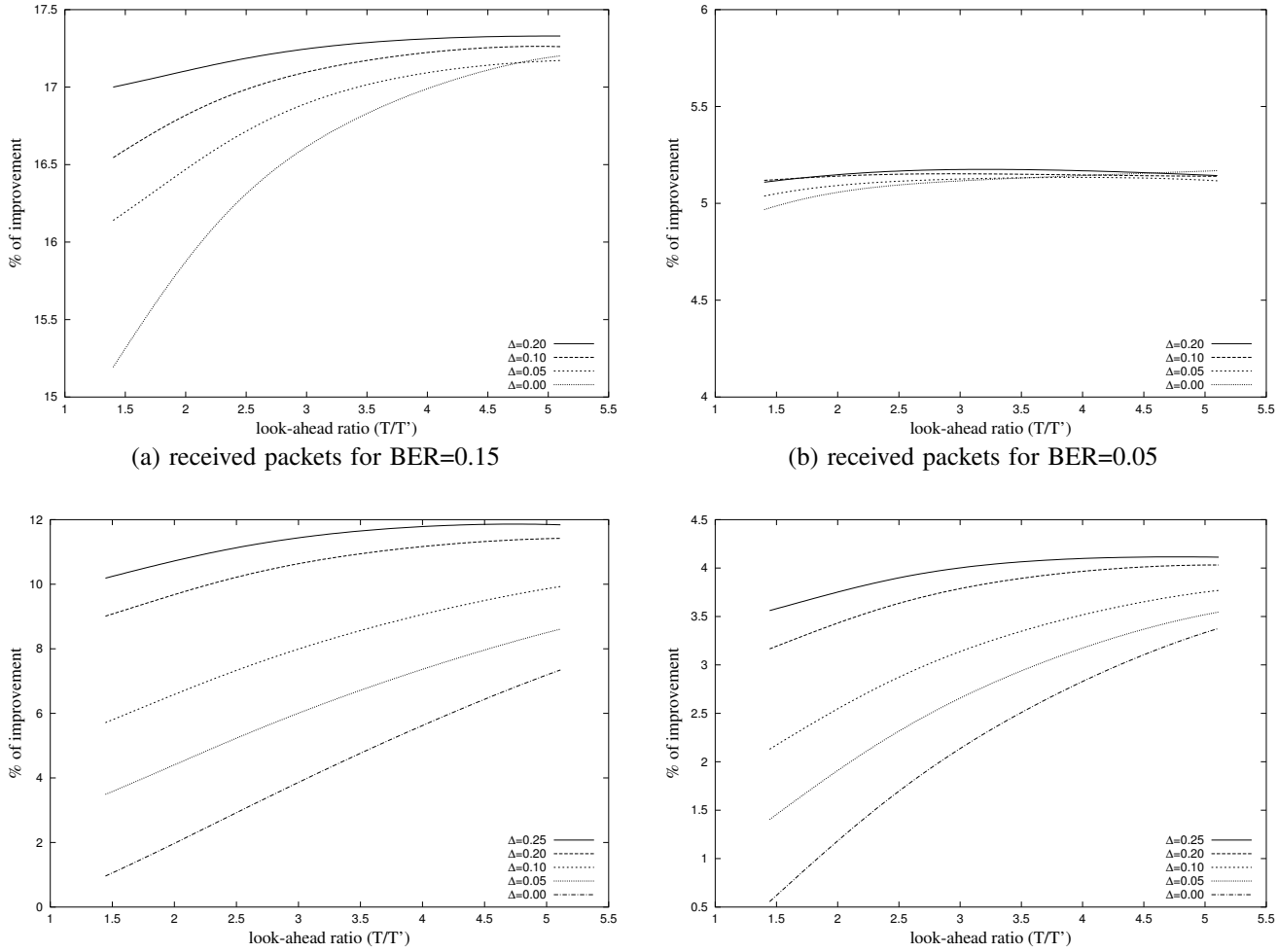
The synchronous application we start with is a video over IP codec that generates data samples at a rate of 256 Kb/s, with packetization time of 40 ms. Consequently, each packet contains 1340 bytes (32KB/s*0.04) of video samples and 60 bytes of headers. The considered tolerated jitter is 1 sec. The total bandwidth required by all the active video calls is 30% of the entire channel bandwidth. Figure 8 depicts the reduction in the bandwidth consumed by the quantitative algorithm vs. the reference algorithm, as a function of the "error burst length." The latter value represents the average time period during which the channel is in the Gilbert model bad state, and it is determined in our simulations independently of the average error rate. We ran the simulations for several values of average error rates (5%, 10% and 20%).

In Figure 8(a) we consider the case where packet retransmission is not allowed, whereas in Figure 8(b) we consider the case where one retransmission at the MAC layer is allowed. It is evident that in both cases the contribution of our algorithm is greater when the average bit error rate (BER) is higher. For example, when the error burst length is 0.5 sec. and the BER is 10%, our algorithm reduces the bandwidth consumed by the video application by 2.5% when only one copy of each packet can be transmitted, and by 8% when a packet can be retransmitted exactly once.

We discussed in Section III several approaches for avoiding transmission in a bad channel due to a relatively short scheduler lookahead interval. One was to impose a minimum threshold Δ on the probability of a successful transmission. This threshold is one of the most important parameters in the quantitative-based algorithm. It determines how aggressive the algorithm is when channel conditions are poor. The results shown in Figure 8 were achieved for $\Delta = 0.05$. Although not shown in this figure, the number of delivered packets was almost equal (the differences are in the order of 0.1%).

In Figure 9 we compare the use of a minimum threshold Δ to the use of a high T'/T ("lookahead ratio") value, where $T'/T = 1$ indicates no "lookahead", as in Figure 8. We still consider video connections, but this time the tolerated jitter is 5 sec. The load imposed by the synchronous traffic is 30% of the channel bandwidth, and each packet can be transmitted at most twice (one retransmission). The average length of an error burst is 0.5 sec. and the average BER is 0.05 and 0.15.

Consider Figure 9(a) first. This figure shows the reduction in the bandwidth consumed by the quantitative algorithm vs. the reference algorithm as a function of T'/T, for several values of Δ. When T'/T increases, the quantitative algorithm improves further in comparison to the reference algorithm. However, this improvement is stable when $T'/T \geq 5$. It is also evident that with this lookahead there is no need to use the Δ threshold because the performance does not change when $\Delta = 0$. It is interesting to note that when the lookahead ratio is small, we can increase the number of correctly received packets by increasing the value of Δ. The reason is that in this model we allow each packet to be transmitted at most twice. Therefore, by avoiding transmissions when the condition of the channel



(a) received packets for BER=0.15

(b) received packets for BER=0.05

(c) transmitted packets for BER=0.15

(d) transmitted packets for BER=0.05

Fig. 9. The performance as a function of Δ and lookahead ratio T'/T when synchronous load is 30%

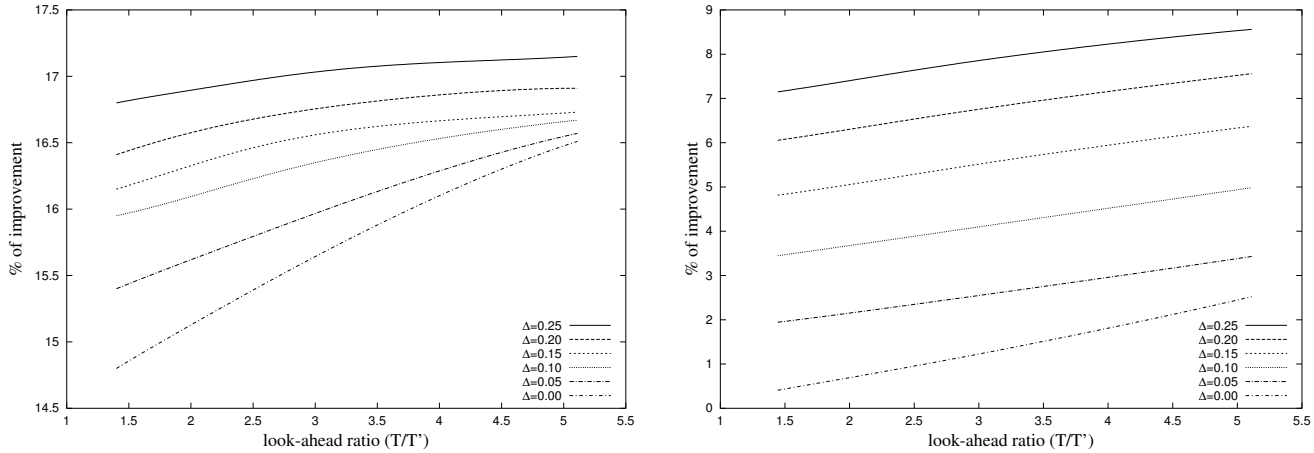
is likely to be bad, we actually increase the probability of a packet to be correctly received.

Figure 9(b) is similar to Figure 9(a) except that the BER is 0.05 rather than 0.15. We see that the quantitative algorithm increases the number of successfully received packets by 5% regardless of the value of T'/T and Δ. The main reason for the constant improvement is that, with the aforementioned average loss rate, length of error burst, and tolerated jitter, every transmitted packet is likely to be received either in the first or in the second trial. Indeed, the percentage of delivered packets in this case was found to be very close to 100%.

Next, consider Figure 9(c) and (d). The setting is similar to what we considered in Figure 9(a) and (b) respectively, except that here we show the improvement of the quantitative algorithm from the perspective of the number of transmitted packets. An improvement of 10% here (e.g., when T'/T=1.5 and $\Delta = 0.25$) indicates that the quantitative algorithm transmits only 90% of the packets transmitted by the reference algorithm. It is evident that a relatively high value of Δ can compensate for a low value of T'/T. This is because the tolerated jitter length in this case is much longer than the average error burst length (5 vs. 0.5 sec.). Therefore, when T'/T is small but Δ is high, the scheduler does not transmit a packet in a bad channel, but rather waits until the channel becomes good. Moreover, such high values of Δ do not affect the number of correctly received packets. If the tolerated jitter length were shorter, the increase of Δ beyond a certain point would reduce the number of successfully received packets.

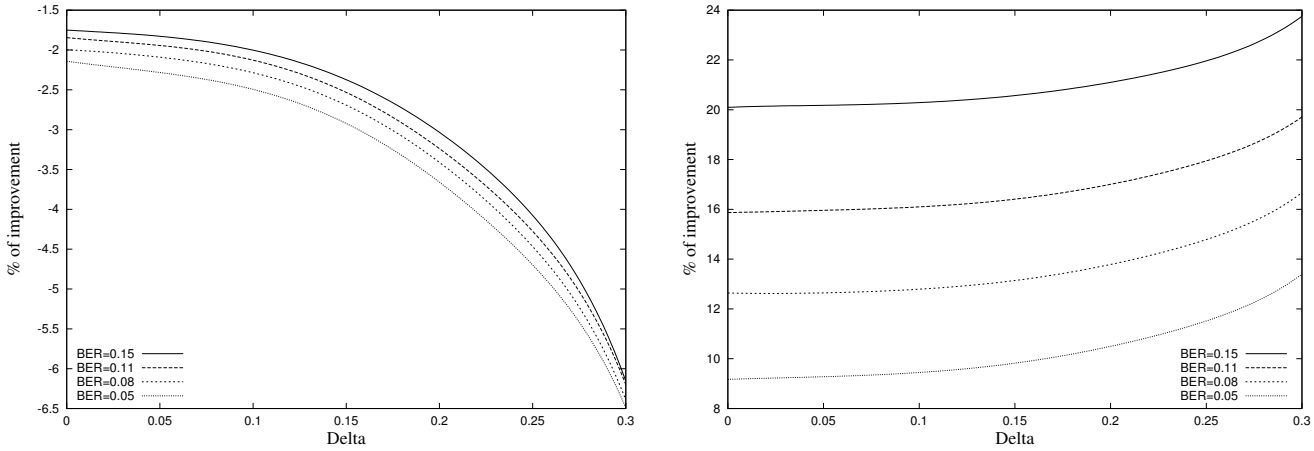
Recall that in the model considered above the load imposed by the synchronous traffic was 30%. In addition, we assumed that 10% of the bandwidth is consumed by best-effort applications. When this traffic increases, the performance of our algorithm improves, because the scheduler has more accurate information about the status of each active channel.

The model considered in Figure 10 is similar to the one considered in Figure 9 except that the load imposed by the synchronous traffic increases from 30% to 60% of the channel bandwidth. We increased the synchronous traffic by activating more hosts and not by increasing the load imposed by each active host. This implies that the scheduler does not acquire more accurate information regarding the status of the active uplink channels. Consequently, and since when the load is



(a) received packets (b) transmitted packets

Fig. 10. The performance as a function of Δ and lookahead ratio T'/T when synchronous load is 60%



(a) received packets (b) transmitted packets

Fig. 11. The performance as a function of Δ and BER for Scenario C

higher it is more difficult for the scheduler to find profitable slots for each pending packet, the normalized net profit of the scheduler (not shown in the graph) is smaller than in the previous case. In Figure 10 we show only the case where the average BER is 0.15. As before, the y-axis indicates the percentage of improvement over the reference algorithm. We can see that in terms of the number of correctly received packets, our algorithm is still better by $\sim$ 17% than the reference algorithm (that is, 17% more received packets). In terms of the number of transmitted packets, our algorithm is better by $\sim$ 7% with no lookahead and by $\sim$ 8% with lookahead ratio of 5.

The last graph we show (Figure 11) is for Scenario C, namely, for the case where the tolerated jitter is shorter that the error burst length. This is a typical scenario for mobile users activating voice-over-IP synchronous calls. In the considered simulation model the average error burst length is 500 ms., the tolerated jitter is 10 ms. and the load of the synchronous traffic is 0.8. Due to the small jitter value, working here with $T'/T > 1$ does not improve the throughput. As discussed in Section I and outlined in Figure 1, the main challenge of an efficient scheduler in Scenario C is to minimize the number of bad synchronous transmissions. We compare again the results of the scheduler presented in this paper to the results of the reference scheduler discussed earlier in this section. As shown in Figure 11(b), our scheduling algorithm decreases the number of transmitted packets by 8-24%, depending on the BER and the value of Δ. However, with respect to the number of correctly received packets (Figure 11(a)), the improvement is negative: our algorithm delivers fewer packets than the reference algorithm. For example, when the BER is 0.15 and $\Delta = 0.15$, it transmits 20% fewer packets, and delivers 3% fewer packets. This model reveals that using a high threshold Δ is a double-edged sword: it helps reduce the number of transmitted packets while (slightly) decreasing the number of those received.

We have shown throughout this section that the proposed algorithm is significantly better than the reference algorithm for many different cases and scenarios. We conclude this section with two observations regarding the recommended

values of Δ and lookahead ratio T'/T:

- The performance of the algorithm, both in terms of fewer transmissions and in terms of more successfully received packets, is significantly improved for synchronous calls whose tolerated jitter is $\gg T$ when a lookahead ratio of $T'/T > 1$ is used. However, this improvement stops when a value higher than 5 is used. Hence, the recommended value of T'/T is 5.
- When the tolerated jitter is $\gg T$, a relatively large value of Δ (~ 0.3) improves the performance. However, such a value can reduce the number of successfully received packets for synchronous calls whose tolerated jitter is short. Hence, we recommend working concurrently with multiple values of Δ: for example, $\Delta \approx 0.1$ for calls whose tolerated jitter is short (tens of milliseconds), $\Delta \approx 0.2$ for calls whose tolerated jitter is medium (hundreds of milliseconds), and $\Delta \approx 0.3$ for calls whose tolerated jitter is relatively long (several seconds).

VII. Conclusions

In this paper we presented a generic quantitative-based scheme for scheduling the transmission of synchronous packets over a wireless access channel. We identified three scheduling scenarios with which a generic algorithm has to cope, and showed that the scheduler logic has a different challenge when addressing each of them. The proposed generic scheduling algorithm translates all the factors relevant to each scenario into a common profit parameter, and selects the most profitable transmission instances.

The benefit of the proposed scheduling algorithm is threefold: (a) it selects the most important packets for transmission; (b) it increases the number of synchronous packets that are transmitted on time, and (c) it decreases the number of packets that are transmitted when the channel is noisy.

We showed how the scheduler translates the status of the channel into a profit metric both for static and for mobile nodes. This metric can reflect the robustness and bandwidth cost of each possible PHY profile, and it can also account for possible future retransmissions, when applicable. We presented approximation algorithms that allow the scheduler to select the most profitable transmission instances during each scheduling period. Finally, we used simulations in order to understand how the various parameters affect the performance of the proposed scheduler in several cases, and in order to compare this performance to the performance of a reference algorithm.

References

[1] M. Adamou, S. Khanna, I. Lee, I. Shin, and S. Zhou. Fair real-time traffic scheduling over a wireless LAN. In *Proc. of the 22nd Real-Time Systems Symp.*, December 2001.

[2] Aytac Azgin and Marwan Krunz. Scheduling in wireless cellular networks under probabilistic channel information. In *ICCCN*, pages 89–94, 2003.

[3] C. Chekuri and S. Khanna. A PTAS for the multiple knapsack problem. In *The 11'th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 213–222, 2000.

[4] R. Cohen, L. Katzir, and D. Raz. Scheduling algorithms for a cache pre-filling content distribution network. In *INFOCOM'2002, NYC, NY*, June 2002.

[5] Reuven Cohen and Liran Katzir. A generic quantitative approach to the scheduling of synchronous packets in a shared medium wireless access network. In *INFOCOM*, 2004.

[6] C. Eklund. IEEE standard 802.16: A technical overview of the wireless MAN air interface for broadband wireless access. *IEEE Communications Magazine*, June 2002.

[7] A. J. Goldsmith and S. G. Chua. Variable-rate variable-power MQAM for fading channel. *IEEE Transactions on Communications*, 45(10), October 1977.

[8] Institute of Electrical and Electronics Engineers Inc. IEEE Standard for Local and Metropolitan Area Networks – Part 16: Air Interface for Fixed Broadband Wireless Access Systems, 2004.

[9] Institute of Electrical and Electronics Engineers Inc. IEEE Draft Standard for Local and Metropolitan Area Networks – Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, April 2005.

[10] Didem Kivanc, Guoqing Li, and Hui Liu. Computationally efficient bandwidth allocation and power control for OFDMA. *IEEE Transactions on Wireless Communications*, 2(6), November 2003.

[11] J. Lemmon. Wireless link statistical bit error model. Technical Report 02-394, U.S. Dep. of Commerce, June 2002.

[12] S. Lu, V. Bharghavan, and R. Srikant. Fair scheduling in wireless packet networks. *IEEE/ACM Transactions on Networking*, 7(4):473–489, 1999.

[13] S. Nanda, K. Balachandran, and S. Kumar. Adaptation techniques in wireless packet data services. *IEEE Communications Magazine*, 38(1), January 2000.

[14] H. Sanneck, N. Le, M. Haardt, and W. Mohr. Selective packet prioritization for wireless VoIP. In *4th International Symposium on Wireless Personal Multimedia Communication*, September 2001.

[15] H. Sanneck, N. Le, A. Wolisz, and G. Carle. Intra-flow loss recovery and control for VoIP. In *ACM Multimedia*, September 2001.

[16] S. Shakkottai and R. Srikant. Scheduling real-time traffic with deadlines over a wireless channel. *ACM/Baltzer Wireless Networks Journal*, 8(1):13–26, January 2002.

[17] G. Song and Y. Li. Cross-layer optimization for OFDM wireless networks, part I: Theoretical framework. *IEEE Transactions on Wireless Communications*, 4(2), March 2005.

[18] D. Tse and S. Hanly. Multi-access fading channels: Part I: Polymatroid structure, optimal resource allocation and throughput capacities. *IEEE Transactions on Information Theory*, 44(7):2796–2815, November 1998.

[19] P. Viswanath, D. Tse, and R. Laroia. Opportunistic beamforming using dumb antennas. *IEEE Transactions on Information Theory*, 48(6), June 2002.

[20] M. Zorzi, R. Rao, and L. B. Milstein. On the accuracy of a first-order Markov model for data transmission on fading channels. In *IEEE ICUPC'95*, November 1995.