To appear in Proc. 10th IEEE Intl. Conf. on Emerging Technologies and Factory Automation, ETFA 2005, Catania/Italy

Scheduling Multiple Streams with (m, k)-firm Deadlines having Different Importance over Markovian Channels

Andreas Willig
Telecommunication Networks Group (TKN) at Technical University Berlin
Sekr. FT-5
Einsteinufer 25
10587 Berlin, Germany
awillig@ieee.org

Abstract

There are several (not only industrial) applications which can tolerate some losses. A concise way to distinguish between allowed and forbidden loss patterns are (m, k)-firm deadlines, which state that at most m out of k consecutive packets may get lost. An obstacle to the fulfillment of even the relaxed (m, k)-firm deadlines are channel errors. In this paper we consider the case where multiple periodic streams having the same period and their own (m_i, k_i)-firm deadlines are to be scheduled over a common, error-prone communications link. For each new packet transmission a scheduler has to decide which stream to serve next. A number of scheduling policies is investigated over a range of simple channel models of varying burstiness. It shows up that already in this simple case none of the scheduling policies is best (in the sense of having the least costs incurred when violating (m_i, k_i)-firm deadlines) for all different values of the burstiness, but the best policy depends on the actual channel burstiness. One approach to improve this would be to design decision rules like "if burstiness is so and so, use policy A, otherwise use policy B". However, since it cannot be expected that such rules carry over to more elaborate channels, we have designed an adaptive "meta-scheduler", which continuously selects out of a set of candidate policies a policy that might be the best in the current channel situation because it would have been the best one in the immediate past. Depending on its parameters, the meta-scheduler achieves close-to-optimal performance.

1. Introduction

In the design of wireless fieldbus systems the presence of channel errors should be carefully considered in the design of protocols, in the formulation of quality-of-service requirements and in the dimensioning of the system [11, 1]. After accepting that it is hardly possible to give *deterministic guarantees* for worst-case packet transmission times or perfect delivery reliability on wireless channels, applications and protocols can be designed such that they can deal with "some losses". The quite imprecise notion of "some losses" means that there are loss patterns with which the application can live, but other loss patterns are unacceptable. Still, the unpredictable nature of transient wireless channel errors might lead to situations where unacceptable loss patterns occur, and insight is needed how often this will happen.

In this paper, we consider a certain way to disambiguate allowed from forbidden loss patterns: The concept of (m, k)-firm deadlines [5, 6]. We explain this in the context of a digital controller which requests periodically data from a sensor. The numbers m and k are integers such that $m < k$, and the (m, k)-firm deadline is met when at most m out of a window of k *consecutive* sensor data packets have not reached the destination or were not transmitted within their deadline.[1] The traditional deterministic requirement that all deadlines must be met can be expressed as an $(0, 1)$-firm deadline. If an (m, k)-firm deadline is not met, we say that a *violation* occurs. If a violation of this constraint is critical, the system is called a *weakly hard real-time* system [8].

In this paper we consider a controller that polls multiple sensors. All packets have the same size, the time is slotted and a time slot accommodates the controllers poll packet as well as the sensors answer data. Each sensor generates a *stream* of packets, i.e. at every time slot a new packet is created by the sensor.[2] For each stream $i \in \{1, \ldots, M\}$ there is a separate (m_i, k_i)-firm deadline. The controller has only a single network adapter, which shares a single, lossy medium with the sensors. Whenever the controller wants to poll a sensor, it has to decide which of the sensors/streams is to be polled – it has to make a

[1] The formulation in [5, 6] states that a minimum of m packets must have *reached* the controller to meet the (m, k)-firm deadline. Clearly, both formulations are equivalent.

[2] This means that all streams have the same period. While this assumption is often not true in practice, it serves as a worst-case assumption in this paper.

scheduling decision. We assume that to each stream i a cost $C_i > 0$ is associated, which is incurred each time the stream experiences a violation. The optimization goal for the scheduling algorithms pursued in this paper is to minimize the per-slot average deadline violation costs incurred by the scheduler.

One important obstacle are channel errors. This is especially true when wireless channels are considered, which are known to have time-varying and sometimes quite high error rates [10]. It is for many reasons more practical to work with artificial stochastic models of wireless channels instead of doing measurements: measurements in wireless channels are often not reproducible, they are hard to set up, and measured traces can be quite huge and clumsy to deal with. In stochastic channel models, the channel behaves according to a prescribed type of stochastic process, which relies on a (typically small) number of parameters. One versatile and often-used class of channel models for wireless channels are hidden Markov models [7], which includes as a special case the Gilbert-Elliot (GE) model [4, 2].[3] In this paper we restrict to the GE model, the consideration of more general HMM channels is subject of future work.

This paper is a first step in developing a deeper understanding of the interplay between scheduling policies and the behavior of wireless channels and in the search for good schedulers. Specifically, the contributions of this paper are the following:

- It is shown that the channel error patterns have a significant impact on the violation costs. In our specific example, a single stream having quite relaxed requirements (a $(5-10)$-firm deadline) is investigated over a number of channels having the same long-term average packet error rate of 10%. It is shown that the violation probability depends on the "burstiness" of the channel, which roughly describes the length of good and bad channel periods.

- A number of different scheduling policies is investigated over a range of channels (having all the same average packet error rate of 10%), and it is demonstrated that the optimal policy depends on the channel error patterns.

- An adaptive scheduling policy is presented, which lets the scheduler select out of a fixed set of given ones (*candidate policies*) a single policy which would have been optimal in the immediate past. An advantage of this scheme is that it does not rely on pre-formulated heuristics of the kind "when the burstiness is so and so, scheme A might be the best one, otherwise use scheme B", only a history of transmission outcomes is required. Developing such heuristics for the channels considered in this paper is not a problem in itself, but it is far from obvious whether they would apply for other channels than those considered in this paper.

[3] In Hidden-Markov models (HMM) the probabilistic behaviour is governed by a Markov chain with some state space I, a finite output alphabet $\{\alpha_1, \ldots, \alpha_o\}$ and for each $i \in I$ a probability measure $p_i(\cdot)$ on the output alphabet. When the Markov chain switches into a new state i, the next output symbol is independently drawn from the output alphabet according to the probability distribution $p_i(\cdot)$. In the context of channel models, the Markov chain state space I models different "channel regimes", the output alphabet contains the elements 0 (denoting "no bit error") and 1 (denoting "bit error"). With such a model we can assign to every channel regime an own bit error rate.

This paper is structured as follows: in the following Section 2 we describe the system under consideration more precisely. In Section 3 we give a numerical example showing that even for a single stream the violation probability does not only depend on the average packet loss rate, but on further channel attributes. In Section 4 different scheduling policies are investigated for their average per-slot violation costs when serving a number of streams. The next Section 5 introduces an adaptive scheme, which lets the base station decide on the actual scheduling policy to use, based on a history of transmission outcomes. Conclusions are given in Section 6.

Some analytical results are just quoted in this paper, the details can be found in a technical report [9].

2. System Model

We are given a *base station*, a number M of *slaves* or *sensors*, and a channel between them. The base station polls the slaves, which send answer data back. Each slave generates new answer data upon every time slot. These data packets can be either transmitted correctly or the channel introduces errors which are always detected at the base station. The base station drops erroneous data packets. For simplicity it is assumed that poll packets are always transmitted correctly.

The time is slotted, the length of a time slot is sufficient to accommodate the poll packet and the answer data packet; all data packets have the same size. The time instants $\tau_0, \tau_1, \tau_2, \ldots$ correspond to the start times of the first, second, third, ... time slots. The channel is governed in the general HMM case by a finite and time-homogeneous discrete-time Markov chain (TH-DTMC) with transition matrix $\mathbf{P}$. The N channel states are $\{0, 1, \ldots, N-1\}$, and to each state a packet error probability $P_i \in [0, 1]$ is associated. We assume that the Markov chain is irreducible and aperiodic and thus has a steady-state vector. The precise operation of such HMM channels is as follows: When the channel is in state i at the beginning of the n-th packet transmission (i.e. at time τ_{n-1}), then the packet is transmitted correctly with probability $1 - P_i$ and is erroneous with probability P_i. Packet transmissions in different slots are independent. At the end of this time slot, the channel transitions from state i into a new state j with probability $p_{i,j} = [[\mathbf{P}]]_{i,j}$, this transition being independent of the packet outcome and of previous state transitions.

It is assumed that each sensor $s \in \{1, \ldots, M\}$ generates a packet stream with its own (m_s, k_s)-firm deadline

and with a period corresponding to the slot time – this is a worst-case assumption regarding the system load. At the beginning of a time slot the base station's scheduler selects one out of the M sensors/streams for service and sends a poll packet to this sensor. The base station maintains a number t_s describing the state of sensor s. To ease notation, we fix on a specific stream s and denote by t its state, and by m and k the numbers m_s and k_s, respectively. The number t is in the range between 0 and $2^k - 1$. The binary representation of the state is given as:

$$t = \nu_{k-1}2^{k-1} + \nu_{k-2}2^{k-2} + \ldots + \nu_0 \qquad (\nu_i \in \{0,1\})$$

In this representation, ν_0 represents the outcome of the last transmission, ν_1 the outcome of the next-to-last transmission, and so on. A positive outcome ($\nu_0 = 0$) occurs when stream s has been selected by the scheduler for service *and* the data packet has been received successfully by the base station. In all other cases (stream s has not been selected by scheduler, selected but transmission failure), the outcome is negative ($\nu_0 = 1$). We specify the stream state t interchangeably as the above number as well as by the vector $(\nu_{k-1}, \ldots, \nu_0)$. Accordingly, the "one-norm"

$$\|t\|_{k,1} := \sum_{j=0}^{k-1} \nu_j$$

indicates how many of the last k transmission attempts were in error. If this sum is larger than m, then a *violation* occurs and the costs C_s are incurred.

The base station performs the following actions at the beginning of a time slot:

- It selects one of the M streams/sensors for service, its state variables being t, m and k.
- It sends a poll packet to the sensor at time τ_n and expects the answer data packet.
- If the packet is received successfully, the new state t' of stream s at the $n+1$'st time slot is updated as:

  $$t' = \nu_{k-2}2^{k-1} + \nu_{k-3}2^{k-2} + \ldots + \nu_0 2^1 + 0$$

  whereas in case of a transmission error the new state is

  $$t' = \nu_{k-2}2^{k-1} + \nu_{k-3}2^{k-2} + \ldots + \nu_0 2^1 + 1$$

  Stated differently: the new outcome is "shifted by one from the right" into the state variable, which we write symbolically as:

  $$t' = ((t \,\mathrm{SHR}\, 1) + o) \mod 2^k$$

  with the outcome o being in $o \in \{0,1\}$.
- For all other streams a one is shifted from the right, since they have not been selected.

From now on we simply speak of a failure for a stream, if it is either chosen for transmission and experiences a true transmission error, or if it is not chosen for transmission.

It is the goal of the scheduler to minimize the long-term average per-slot violation costs. Before we explain the scheduling policies, we introduce the notion of the (time-dependent) *distance* $d(i)$ of a stream i being in state t_i to a violation state: $d(i) = m_i - \|t_i\|_{k_i,1}$. Please note that negative distances mean that the stream is already in violation state. The following scheduling policies have been investigated:

- *closest-to-violation-random* (CTV-R): this policy selects the stream having the smallest distance to a violation. Ties are broken by a random choice.[4]
- *closest-to-violation-highest-cost* (CTV-HC): similar to CTV-R, but ties are broken based on selecting the highest-cost stream, i.e. the stream having the maximum C_i value. Further ties are broken by a random choice.
- *round-robin* (RR): the scheduler selects the streams in a round-robin manner, no matter what their costs or their distance to violation are.
- *prioritize-highest-costs* (PHC): the scheduler selects the highest-cost for which the state t is nonzero, i.e. which actually experienced at least failure during the last k_i packets.
- *linear (w_d, w_c)* (L-(w_d, w_c)): for each stream i a modified distance $d'(i)$ and its cost C_i are used to compute a weighted average (using weights w_c and w_d for cost and distance, respectively), and the stream having the maximum value is returned (ties broken by random choice). The scheduler selects thus the stream

  $$\mathrm{argmax}_{i\in\{1,\ldots,M\}} \left(w_c \cdot C_i + \frac{w_d}{d'(i)} \right)$$

  where $d'(i) = d(i)$ if $d(i) > 0$ and $d'(i) = d(i) - 1$ otherwise. A pair of good weights has been obtained by exploratory simulations using the channel models described below. The strategy L-$(-2, 1)$ has been most often the best one (out of the set of strategies defined by $w_c \in \{0,1,2,3,4\}$ and $w_d \in \{1,0,-1,-2,-3,-4,-5\}$) and always been close to the optimum, therefore, in the following we concentrate on L-$(-2, 1)$.

To avoid visual overloading of the figures in subsequent sections, we state here that the RR and PHC perform poorly and are not included in the figures. The full set of results is contained in [9].

[4] This is the policy closest in spirit to the *distance-based priority* (DBP) scheme presented in [5]. The difference between CTV-R and DBP lies in the fact that in DBP all negative values for the distance are collapsed to a distance of zero. Hence, DBP makes no distinction between streams that have just experienced a violation the first time and streams that are within violation state since long time.

3. Numerical example: a single stream over different Gilbert-Elliot channels

For the case of a single stream we have determined an analytical solution for the probability of a violation over general Hidden Markov channels (see [9]). We evaluate this model numerically for a specific example. We consider a stream having an $(5, 10)$-firm deadline over various instances of a Gilbert-Elliott (GE) channel model [4, 2]. The GE channel model can be defined on the level of packets and allows to introduce short-term correlation in the error process. The channel is modeled with two states, named "good" and "bad", corresponding to states 0 and 1. Within the good state the packet error rate is assumed to be $P_0 = 0$, within the bad state the packet error rate is given by $P_1 > 0$. Specifically, for the GE model we have:

$$\mathbf{P} = \begin{pmatrix} p_{g,g} & 1 - p_{g,g} \\ 1 - p_{b,b} & p_{b,b} \end{pmatrix}$$

The steady-state vector $\pi = (\pi_0, \pi_1)$ of $\mathbf{P}$ is given by:

$$\pi_0 = \frac{1 - p_{b,b}}{2 - (p_{g,g} + p_{b,b})} \ , \ \pi_1 = \frac{1 - p_{g,g}}{2 - (p_{g,g} + p_{b,b})}$$

and the average steady-state packet error rate is given by:

$$P^* = P_0 \cdot \pi_0 + P_1 \cdot \pi_1 = P_1 \cdot \pi_1$$

The state holding times are geometrically distributed. The average state holding times (measured as number of slots) for the good state $E[H_0]$, and the average state holding time for the bad state $E[H_1]$, are given by:

$$E[H_0] = \frac{1}{1 - p_{g,g}} \qquad E[H_1] = \frac{1}{1 - p_{b,b}}$$

We fix the average packet error rate as $P^* = 0.1$, i.e. the packet error rate is 10%. For a single experiment also $E[H_1]$ is kept fixed, and $p_{b,b}$ is computed from this. The parameter varied within an experiment is the *burstiness index* B defined as $B = E[H_1]/E[H_0]$, i.e. as the ratio of the average bad state holding times to the average good state holding time. For given B and $E[H_1]$, one can solve immediately for $E[H_0]$, and in a second step for $p_{g,g}$. Having this, it is possible to compute P_1 such that $\pi_1 \cdot P_1 = P^*$. The burstiness index cannot become arbitrarily small, because of the constraint $0 < P_1 \leq 1$. Too small values of B would lead to $P_1 > 1$, which is impossible.

In Figure 1 we show for different values of the average bad state holding time ($E[H_1] \in \{5, 10, 15, 20, 25\}$) and for varying burstiness index the violation probability of a $(5, 10)$-firm deadline. The following points are remarkable:

- The violation probability is sensitive to both the average bad state holding time $E[H_1]$ as well as the burstiness index, even when all channels have the same average packet error rate $P^* = 0.1$.

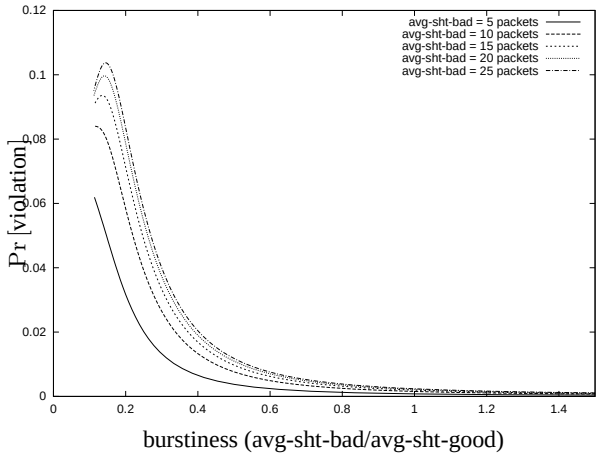

Figure 1. Numerical example for an $(5, 10)$-firm deadline. Different variations of a Gilbert-Elliot channel are used. The burstiness index is defined as the ratio of the average state holding times in the bad state divided by the avg. state holding time in the good state. Curves for different avg. bad state holding times are shown.

- All curves have their highest violation probabilities for small burstiness indices. In this situation the good state holding times are large, and to achieve the given average packet error rate $P^* = 0.1$, the packet error probability in bad state P_1 becomes close to one. Accordingly, as soon as the channel enters the bad state, the $(5, 10)$-firm deadline can be easily violated, especially when the average bad state holding times are larger than $m = 5$.

This demonstrates that the violation probability is influenced not only by the long-term average packet error rate, but also by the presence (and amount of) short-term correlation in the channel error patterns.

4. Simulation example: multiple streams over different Gilbert-Elliot channels

In this experiment we consider a set of $M = 3$ streams. Stream one has a $(m_1, k_1) = (3, 4)$-firm deadline and violation costs of $C_1 = 4$. Stream two has a $(m_2, k_2) = (5, 8)$-firm deadline and violation costs of $C_2 = 2$. Finally, for stream three we have $(m_3, k_3) = (8, 10)$ and $C_3 = 1$. This assignment of (m_i, k_i)-firm deadlines allows to have up to 20% packet loss on the channel without deadline violation (remember: the average loss rate P^* is set to 10%), although not in arbitrary time slots. The long-term average per-slot costs for this set of streams and for the different scheduling policies is evaluated by simulation over the same set of Gilbert-Elliot channels that has been used in the last section. A single simulation run

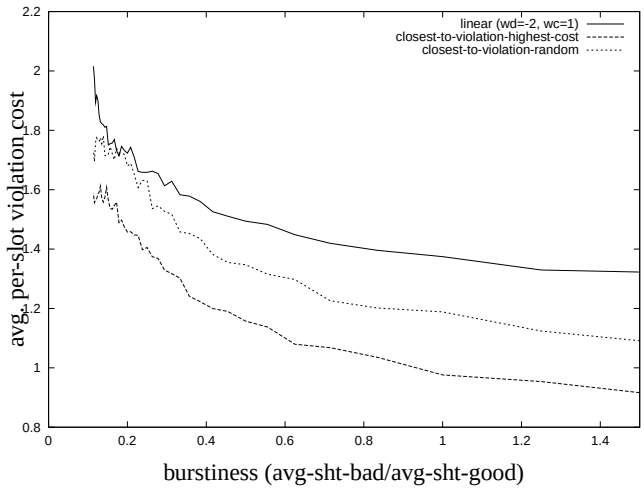

Figure 2. Avg. per-slot violation costs for the different scheduling policies vs. burstiness index. The average bad state holding time is $E[H_1] = 5$ packets.

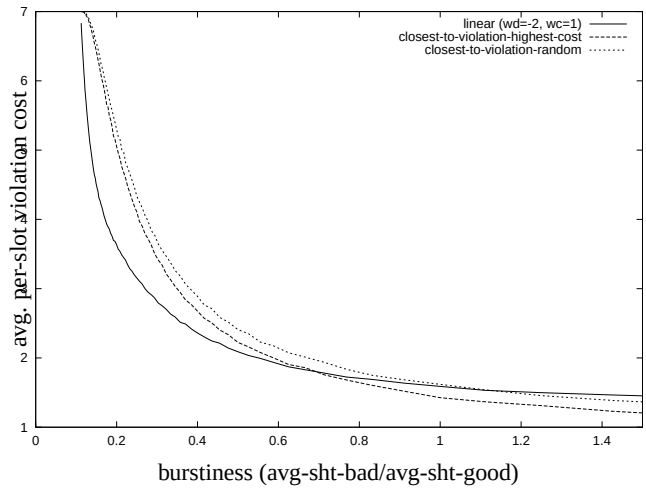

Figure 3. Avg. per-slot violation costs for the different scheduling policies vs. burstiness index. The average bad state holding time is $E[H_1] = 10$ packets.

spans a number of 100,000 packets, and the average costs per slot are simply obtained by dividing the overall accumulated costs by the number of packets in the simulation.

In Figures 2 and 3 we show the average per-slot violation costs for the different scheduling policies vs. the burstiness index for average bad state holding times of $E[H_1] = 5$ and $E[H_1] = 10$, respectively. The curves for $E[H_1] = 15$, $E[H_1] = 20$ and $E[H_1] = 25$ look very similar to Figure 3 and are not shown here. The following points are remarkable:

- The optimal policy depends on the channel, in our case on the burstiness index. For $E[H_1] = 5$ the CTV-HC policy is the best one (i.e. has the least average per-slot violation costs) for all burstiness indices. For the other values of $E[H_1]$, the best policies are the linear L-$(-2, 1)$ policy for the small burstiness indices, while beyond a certain threshold index (≈ 0.65) the CTV-HC policy is optimal.
- For all the different values of average bad state holding times $E[H_1]$ the policy CTV-HC is always slightly better than CTV-R. This was to be expected because of the lack of cost-awareness of CTV-R.

Hence, the two disciplines CTV-HC and L-$(-2, 1)$ provide good results, but it depends on the channel burstiness (or more general: on channel error characteristics) which one is better. There is good reason to believe that this finding (no single scheme is best under all circumstances) carries over to other kinds of channel models as well.

5. A Meta-Scheduling Scheme

In the previous section we have provided evidence that none of the investigated scheduling policies is the best under all different channel conditions. It would thus be desirable to let the base station instantaneously select the policy that might work best under the current channel conditions. In our example, the scheduler could switch between the CTV-HC and L-$(-2, 1)$ rules, depending on its estimate of the current channel burstiness. It would be straigthforward to formulate decision rules for the simple Gilbert-Elliot channels investigated in this paper, for example a rule like "when the observed burstiness is below 0.6, use the linear policy, otherwise use the CTV-HC policy". However, two problems would have to be solved. The first one is that estimates of the current channel burstiness are needed. It is possible, although not trivial, to design suitable estimators. The second problem is more serious: Even when we have developed such a set of decision rules, it cannot really be expected that they carry over in an unchanged manner to other types of channels, like HMM channels with more than two states or even to channels with self-similar error characteristics. Each decision rule would need to be tested under a wide range of different channel types, and the formulation of these rules might become complicated when many different channels have to be covered by the same set of rules.

To avoid this problem at all, we advocate another approach, which is similar in spirit to the concept of Meta-MAC protocols [3]. Our approach is called *meta-scheduler*. Briefly, in the meta-scheduler approach the base station keeps a *history* of the transmission outcomes in the last h time slots, h is called the *history depth*. The history entries carry no more information than just "failed transmission" or "successful transmission". The meta-scheduler knows a set $\mathcal{P}$ of *candidate policies*, in our case this could be the CTV-HC and L-$(-2, 1)$ policies. Before each new time slot, the meta scheduler selects from the

candidate policies the policy $\pi \in \mathcal{P}$ that *would have incurred the least total costs* over the given history. More precisely: for each candidate policy it is evaluated, which overall costs would have been incurred, when the policy is started with streams in state zero and with transmission outcomes as given by the contents of the history. For the next slot the policy giving the minimum overall costs over the history is selected. The rationale behind this approach is the assumption that a policy, which would have been the best one in the immediate past, will likely be good in the near future as well.

The meta-scheduling approach has the following properties:

- After each new transmission the history changes (the oldest outcome is pushed out and the newest outcome is added), and the candidate policies have to be re-evaluated. Accordingly, the computational overhead increases linearly in both the number of candidate policies as well as the history depth h. Variations of this scheme are possible, e.g. to decide on a new policy only every m-th time slot.

- The choice of history depth is critical, a number of considerations play a role here. The computational requirements favor a small history depth. On the other hand, a too short history could lead to wrong conclusions, since in evaluating a policies performance all streams start "freshly", i.e. in state zero and the initial transient might change results.

- The history depth also decides between stability and agility of the scheduler, i.e. how well it adapts to changing channel conditions. However, this is not explored in this paper.

We have investigated the meta-scheduling policy for a number of different history depths: $h \in \{10, 20, 30, 50\}$. We compare these meta-policies to the L-$(-2, 1)$ and CTV-HC policies for different average bad state holding times $E[H_1]$. Specifically, in Figure 4 we present the curves for $E[H_1] = 5$ slots, and in Figures 5 and 6 we show the results for $E[H_1] = 10$ and $E[H_1] = 25$, respectively. However, only the meta-policies for $h = 10$ and $h = 50$ have been included in the figures to avoid visual cluttering. In Figure 4 the CTV-HC policy is consistently the best one, the meta-policy with $h = 50$ is close to CTV-HC and the meta-policy with $h = 10$ is worse than the meta-policy with $h = 50$ but significantly better than the linear L-$(-2, 1)$ policy is the worst one. In Figures 5 and 6 it can be observed that the curves for the meta-policies are close to the curves for the two other policies, and for burstiness values $B < 0.4$ and for values $B > 1.1$ the meta-policies are truly in the middle between the other curves. Furthermore, the curve for the meta-policy with $h = 50$ is consistently better than the curve for $h = 10$. The curves confirm that choosing a higher history depth improves the performance of the meta-scheduling policy,

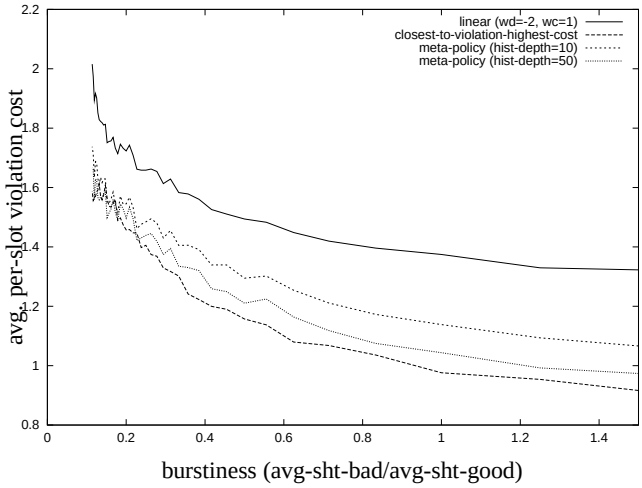

Figure 4. Avg. per-slot violation costs for the different meta-scheduling policies and the L-$(-2, 1)$ and CTV-HC policies vs. burstiness index. The average bad state holding time is $E[H_1] = 5$ packets.

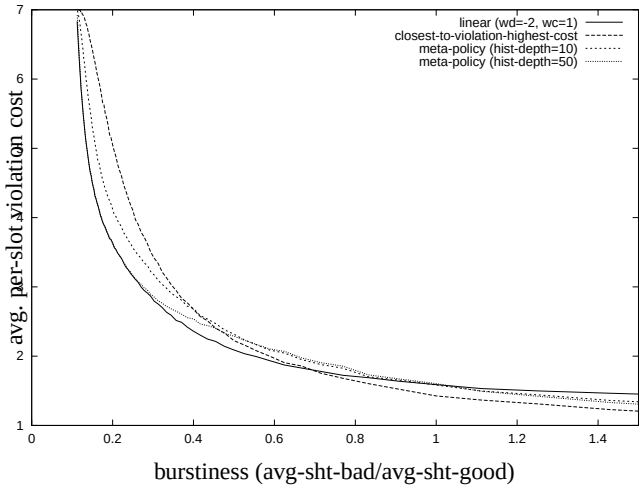

Figure 5. Avg. per-slot violation costs for the different meta-scheduling policies and the L-$(-2, 1)$ and CTV-HC policies vs. burstiness index. The average bad state holding time is $E[H_1] = 10$ packets.

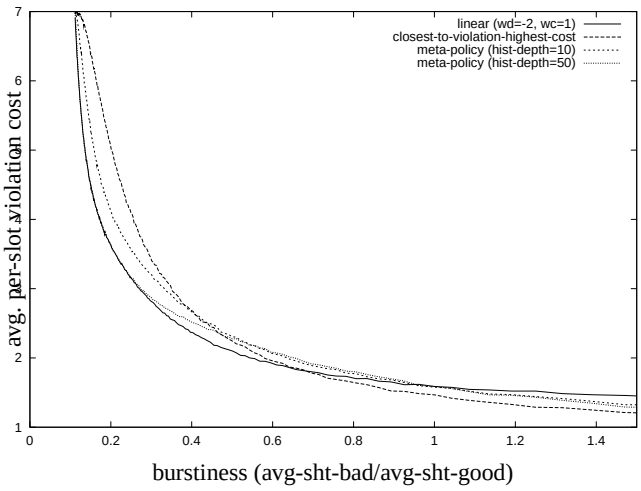

Figure 6. Avg. per-slot violation costs for the different meta-scheduling policies and the L-$(-2,1)$ and CTV-HC policies vs. burstiness index. The average bad state holding time is $E[H_1] = 25$ packets.

for the highest history-depth the meta-policy performs almost as well as the best candidate policy.

In the future we will investigate the performance of the meta-scheduling scheme and the choice of the history depth h for the case of more agile channels. In the case investigated in this paper the channel model is fixed throughout a simulation run. Things might look different if within the same run the channel switches from, say, $E[H_1] = 5$ to, say, $E[H_1] = 15$.

6. Conclusions

The results in this paper represent a first step in the investigation of scheduling policies trying to satisfy (m,k)-firm deadlines of a number of streams over wireless links. The most important result is that none of the investigated scheduling policies is best (i.e. minimizes the average incurred violation costs) under all the different channel conditions investigated in this paper. This insight led to the approach of using a meta-scheduler, which evaluates how each policy out of a set of candidate policies would have behaved in the immediate past, and selects the best-performing policy to make the scheduling decision for the next slot. Running this meta-policy comes at the price of computational costs directly dependent on the number of candidate policies and history depth. However, in our scenario it is the base station who has to pay these costs, which is often more acceptable than putting additional burdens on (possibly energy- or resource-constrained) wireless stations. The meta-policy is a very interesting approach to adapt the behaviour of the base station to time-varying wireless channel conditions.

These results open up lots of opportunities for future research. A first question concerns the choice of the history depth h of the meta-policy scheme, and how this choice relates to channel parameters as well as the k_i values of the different streams. Secondly, to let the set of candidate policies considered by the meta-scheduling algorithm not become too large, good candidate policies need to be identified. Furthermore, it is interesting whether less computationally expensive implementations of the meta-policy scheme than the quite straightforward one used in this paper can be found. Such schemes should avoid iterating over the whole history after every packet. Naturally, the approaches and analyses found in this paper need to be extended to more general HMM channels.

7. Acknowledgements

The author wishes to thank the anonymous reviewers for their insightful comments.

References

[1] J.-D. Decotignie. Wireless fieldbusses – a survey of issues and solutions. In *Proc. 15th IFAC World Congress on Automatic Control (IFAC 2002)*, Barcelona, Spain, 2002.

[2] E. O. Elliot. Estimates of error rates for codes on burst-noise channels. *Bell Systems Technical Journal*, 42:1977–1997, Sept. 1963.

[3] A. Farago, A. D. Myers, V. R. Syrotiuk, and G. V. Zaruba. Meta-MAC Protocols: Automatic Combination of MAC Protocols to Optimize Performance for Unknown Conditions. *IEEE Journal on Selected Areas in Communications*, 18(9):1670–1681, Sept. 2000.

[4] E. N. Gilbert. Capacity of a burst-noise channel. *Bell Systems Technical Journal*, 39:1253–1265, Sept. 1960.

[5] M. Hamdaoui and P. Ramanathan. A dynamic priority assignment technique for streams with (m,k)-firm deadlines. *IEEE Transactions on Computers*, 44(12):1443–1451, Dec. 1995.

[6] M. Hamdaoui and P. Ramanathan. Evaluating dynamic failure probability for streams with (m,k)-firm deadlines. *IEEE Transactions on Computers*, 46(12):1325–1337, Dec. 1997.

[7] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb. 1989.

[8] Z. Wang, Y. qiong Song, E.-M. Poggi, and Y. Sun. Survey of weakly-hard real time schedule theory and its application. In *Proc. International Symposium on Distributed Computing and Applications to Business. Engineering and Science (DCABES)*, 2002.

[9] A. Willig. (m,k)-firm deadlines over markovian channels – (extended version). TKN Technical Report Series TKN-05-003, Telecommunication Networks Group, Technical University Berlin, Apr. 2005.

[10] A. Willig, M. Kubisch, C. Hoene, and A. Wolisz. Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer. *IEEE Transactions on Industrial Electronics*, 49(6):1265–1282, 2002.

[11] A. Willig, K. Matheus, and A. Wolisz. Wireless Technology in Industrial Networks. *Proceedings of the IEEE*, 93(6):1130–1151, 2005.