Large-Deviations Analysis for Energy-Saving Mechanisms in Wireless Networks

Koen De Turck, Dieter Fiems, Stijn De Vuyst, Herwig Bruneel, Sabine Wittevrongel
Department of Telecommunication and Information Processing (TELIN), Ghent University (UGent)
St.-Pietersnieuwstraat 41, B-9000 Ghent, Belgium
kdeturck@telin.ugent.be

ABSTRACT

In wireless telecommunication networks, there is a strong interest in energy efficiency. Sleep-mode mechanisms, which temporarily switch off mobile devices or base stations, are popular means to attain this goal. We look at the performance of different sleep-mode mechanisms with large-deviations tools. These allow us to look at a general multi-user model, for which we obtain a sample path large-deviations principle. As the system equations exhibit discontinuous boundaries, there are some mathematical obstacles to overcome. The results shed light on the trade-off between buffer overflow and energy consumption. We illustrate the results in scenarios with one and multiple mobile stations.

1. INTRODUCTION

Energy-saving mechanisms in wireless communications are currently a hot topic. Short battery life is one of the main impediments to a more widespread use of wireless devices. Hence, understandably, a lot of research is directed at solving or at least mitigating this problem. Evidently, this can be done by improving on battery efficiency, but lately there is also a lot of interest in reducing the energy consumption of the wireless devices by including energy-saving measures in the communication protocols themselves. On that account, it is no wonder that two major standardization efforts for next-generation wireless communications that is, the IEEE-backed 802.16 committee (also known as WiMAX), and the so-called long-term evolution (LTE) of UMTS have opted to incorporate various energy-saving elements which are referred to as 'sleep mode' and 'idle mode' in the WiMAX context and as discontinuous reception (DRX) in LTE terminology. Power saving in WiMAX is achieved by turning off parts of the MS (mobile station) in a controlled manner when there is neither traffic from the MS (uplink traffic) nor to the MS (downlink traffic). Whereas a MS in sleep mode is still registered to a BS (base station) and still performs hand-off procedures, idle mode operation (which is optional in current WiMAX standards) goes further and allows the MS to be completely switched off and unregistered with any BS, while still receiving broadcast traffic. In LTE, similar functionality is present, with user equipment (UE) fulfilling the role of MSs and evolved Node-Bs (eNB) as the BSs.

A considerable number of scientific papers have been devoted to the performance evaluation of sleep-mode mechanisms in the context of wireless networks. A wide variety of tools have been employed: simulations [7] or Markov-chain based (with or without transform-domain methods) [5, 6]. A feature which sets our paper apart is that we look at the impact of sleep mode in a multiple mobile stations scenario. The scenario with multiple mobile stations is too complex to be solved with transform-based or matrix-analytic techniques, therefore we rely on large-deviations tools [**?**, 10], which allow the asymptotic computation of small probabilities on an exponential scale. The rather specific nature of the results (only rare events are considered) is compensated by the broad generality in which the results can be developed. Furthermore, in telecommunication networks, rare events may have a larger impact on performance than 'average behavior', as it is rare events which cause huge delays and subsequently user discomfort and loss of perceived performance.

The model in this paper is not continuous at the boundaries, a condition which complicates large-deviations analyses by a fair amount. Various techniques to work around these discontinuities have been applied, such as the idempotent probability method of Puhalski [8], or the contraction-mapping framework of [10]. In this paper, we opt for an adaptation of the latter (see also [15, 16, 13] for other papers that fit within the same paradigm). Its main idea is the transformation of a large deviation principle of the arrival process into a large-deviations principle of the process of interest by means of the powerful contraction mapping theorem.

Finally, we would like to note that although (the first part of) this article is fairly theoretical in nature, its aim is a practical one: we investigate whether large-deviations techniques can offer quantitative recommendations to network engineers as to how to organize sleep mode in wireless devices. For example, we would like to determine which sleep mechanism is superior from a rare-event overflow probability point of view, the one as defined in the 802.16e protocol [1] or the newly proposed [3]. The examples in Section 5 indeed offer a starting point for such suggestions to the protocol designer.

The outline of the remainder of this paper is as follows. In Section 2, we expound on the modelling assumptions used in this paper. Section 3 contains the main result and its proof. We detail on a numerical computation scheme in Section 4 and illustrate important special cases in Section 5. Finally, we draw conclusions in Section 6.

2. SETTING

Consider a number M of mobile stations (MS) operating under the same base station. As is common in wireless protocols, all operations occur in a slotted (discrete-time) manner. Packets arrive at the base station destined for a specific MS m according to a stationary arrival process $\{\mathbf{X}_m(t)\},\ t \in \mathbb{N}, 1 \leq m \leq M$, and are temporarily buffered at the base station while awaiting transmission, in a dedicated buffer for each MS. In the following, we will assume that the arrival streams $\mathbf{X}_m$ to the different mobile stations are mutually independent but possibly distributed according to a different law. We denote the mean rate of $\mathbf{X}_m$ by λ_m.

For energy-saving purposes, mobile stations may reside in so-called *sleep mode*. The MS then consumes less energy, but is unable to receive packets from the BS. When not in sleep mode, the MS is said to be *active*. We assume that the BS can send packets at a constant maximum rate C which is shared over all active MS. The transmission rate as seen by a single MS is thus greater if fewer stations are active. A number of possible mechanisms of how to organize sleep and active periods in the different MSs have been proposed. We will take a closer look at two of these mechanisms.

S_1: When the buffer dedicated to an MS, say MS m, gets empty, then MS m goes into sleep mode, and a timer is started. The content of the buffer dedicated to traffic for MS m is checked at the end of a series of instants t_i, with service being resumed at the first such instant that the buffer content is non-empty. This is a generalization of the sleep-mode mechanism described in WiMAX. In the WiMAX standard [1], there are three possible sleep-mode classes, of which the most commonly researched is class III, for which the t_i satisfy the following relation:

$$t_i = \max(2^{i-1} t_{\min}, t_{\max}),$$

where $t_{\min}$ and $t_{\max}$ are parameters negotiated between the mobile station and the base station. In LTE, the DRX mechanism is slightly different, resulting in a different sequence of t_i parameters: at first, there is a so-called close-down period, in which the antenna and decoding device is not yet turned off, then a series of short sleep periods of identical length take place, followed by a series of long sleep periods of identical length.

S_2: Sleep and active periods are organized in cycles of a fixed length L. When the buffer of a certain MS m gets empty, it goes into sleep mode for the rest of the ongoing cycle. The technological advantage of this scheme is that it decreases signaling overhead. It has been proposed for the forthcoming IEEE802.16m wireless standard [2, 3]. To the best of our knowledge, no equivalent mechanism has yet been considered for the LTE framework, although it seems as reasonable for this standard as it is in the WiMAX case.

Note that for both mechanisms the system is stable under the natural condition $\sum_m \lambda_m < C$. Before we develop a mathematical formulation of this model, we introduce some notation.

Let $\mathcal{X}$ denote the space of discrete-time $\mathbb{R}^M$-valued processes. Let $\mathbf{X}$ denote a process in $\mathcal{X}$ and its truncation to $[0,t)$ is denoted by $\mathbf{X}[0,t)$. Also, $X(t)$ denotes the value of the process at time t, and $X[0,t) \doteq \sum_{i=0}^{t-1} X_i$, with $X[0,0) = 0$. We are mainly interested in two things: (1) the distribution of the buffer content (and its close relative the packet delay) and (2) the expected energy consumption of an MS station. We develop a set of recursive equations which relate the arrival process to the buffer content process and to the status (active or in sleep mode).

The two mechanisms S_1 and S_2 satisfy the following system equations:

$$Q_m(t+1) = \left(Q_m(t) + X_m(t) - r_m(S_1(t), \cdots, S_M(t))\right)^+. \tag{1}$$

where $\mathbf{Q}_m$ and $\mathbf{X}_m$ denote the buffer content process and the input process of MS m respectively. Also, the auxiliary processes $\mathbf{S}_m, 1 \leq m \leq M$ indicate when MS m is sleeping. The functions r_m determine the rate at which the BS sends to the MS m depending on the state of the other MS. This is a versatile description that allows to model a variety of situations: (1) schemes in which the MS always send at the same rate, regardless of the status of the other MS (no bandwidth redistribution); (2) schemes with bandwidth redistribution. For example, the available bandwidth is divided in equal parts among the active MSs; (3) priority mechanisms where some MS only get bandwidth when other MSs are not working. This setup is of course more general than what we can reasonably explore in Section 5, hence we will only highlight a few scenarios. We just aim to develop a large-deviation result with a natural generality. Note that a setup with no bandwidth redistribution effectively 'uncouples' the queues, so that the single MS case is sufficient for assessing performance.

For S_1 we define another auxiliary process $\tilde{\mathbf{S}}_m$ which counts how long the system has the system been residing consecutively in sleep mode. Its evolution is as follows (we denote by $\vee$ and $\wedge$ denote logical 'or' and 'and' respectively):

$$\tilde{S}_m(t+1) = (\tilde{S}_m(t)+1)\mathbf{1}\left(\tilde{S}_m(t) \notin \mathcal{T} \vee Q_m(t) > 0\right) \tag{2}$$

where $\mathcal{T}$ denotes the set of time instants t_i (i.e. on which the dedicated buffer is checked). The sleep process $\mathbf{S}_m$ for S_1 is then defined as:

$$S_m(t) = \mathbf{1}\left(\tilde{S}_m(t) > 0\right) \tag{3}$$

For sleep mechanism S_2, new cycles start for mobile station m at time instants belonging to the set $L\mathbb{N}+\delta_m \doteq \{Ln+\delta_m : n \in \mathbb{N}\}$ (hence $\delta_m \in \{1, \cdots, M\}$ denotes the offset of MS m). The sleep process $\mathbf{S}_m$ for S_2 satisfies the following recursion:

$$S_m(t+1) = \mathbf{1}\left(Q_m(t) = 0 \vee (t \notin L\mathbb{N} + \delta_m \wedge S_m(t))\right) \tag{4}$$

Finally we make some assumptions on the energy consumption. Each MS consumes a fixed amount E_a during a slot

in which it is active, and an amount E_i during an idle slot. At the end of each sleep interval, an additional amount E_ℓ is consumed (ℓ stands for listening).

3. LARGE DEVIATIONS ANALYSIS

3.1 Motivation

The scenario with multiple mobile stations is too complex to be solved with transform-based or matrix-analytic techniques. Therefore, we opt for a large-deviations analysis. The so-called *many-sources* or *many-flows* scaling [10] yields the most interesting results in this context, and hence we adopt this scaling throughout this paper. We consider a sequence of processes $\mathbf{X}^L$, which are interpreted as the average of L independent processes each distributed like $\mathbf{X}$ and speed up the transmission rate by a factor L as well. The many-flows large-deviations limit describes what happens when the number of flows is very large. In particular, it provides the exponential decay of rare events associated with this limit.

The choice of the many-flows scaling requires some justification. Indeed, in contrast to for example traffic in the backbone of a network, the number of different flows that an individual user sees may not be very large. However, we believe that this modelling choice is justified for the following reasons: (1) Next-generation platforms such as WiMAX aim to operate at a larger, metropolitan area scale, and hence aggregate more traffic; (2) it is one of the few modeling techniques that can still provide answers in the rather general framework of this paper. *Fast-time* scaling, the other frequently used large-deviations scaling is too crude, as it filters away the effects of sleep-mode. Indeed, if the operation of the system is sped up, then sleep periods are also shortened, and vanish in the limit. It is worth noting that if we try to fix this, by scaling the sleep periods with a factor L as we speed up time L times, then we essentially get a simplified many-flows scaling (with no time correlation in the traffic flows).

Let us briefly recall the notion of a large-deviations principle (LDP). We refer the reader to a.o. [9] and [10] for references on large deviations. A sequence of random variables $\mathbf{X}^L$ in a Hausdorff space $\mathcal{X}$ with Borel sigma algebra $\mathcal{B}$ is said to satisfy a large deviations principle with good rate function I if for any $B \in \mathcal{B}$,

$$-\inf_{x\in B^o} I(x) \leq \liminf_{L\to\infty} \frac{1}{L}\log \Pr(\mathbf{X}^L \in B) \leq \limsup_{L\to\infty} \frac{1}{L}\log \Pr(\mathbf{X}^L \in B) \leq -\inf_{x\in \bar{B}} I(x)$$

where $I : \mathcal{X} \to \mathbb{R}\cup\{\infty\}$ has compact level sets, and B^o and $\bar{B}$ denote the interior and closure of the set B respectively. This is called a sample path LDP when $\mathcal{X}$ is a space of processes.

In the following, we will be concerned with the buffer content of one mobile station (say, without loss of generality, MS 1). We denote by q_{S_i}, $i = 1, 2$ two functions which map the arrival processes onto the buffer content of MS 1 at time instant 0 under sleep-mode mechanism S_i. Analogously, mappings $w_{\mathsf{S}_i} : \mathcal{X} \to \mathbb{R}$, $i = 1, 2$ denote the energy consumption of MS 1 during time instant 0.

3.2 Limiting process

It is useful at this point to investigate the behavior of the sleep-mode mechanisms in the many-flows fluid limit. The difference with the large-deviations limit should be clear: whereas the limiting process concerns itself with the almost sure behavior when the number of flows approaches infinity, large deviations aims to quantify the (small) probabilities that the system deviates from this almost sure path. Consider the sequence of processes $\mathbf{Q}_m^L(t)$ which result from plugging $\mathbf{X}_m^L(t) = \frac{1}{L}\sum_{\ell=1}^{L} X_m^{(\ell)}(t)$ into (1), where $\mathbf{X}_m^{(\ell)}, 1 \leq \ell \leq L$ are independent and identically distributed versions of the arrival process.

For many queueing systems, this limiting process (for $L \to \infty$) converges to a deterministic process which moreover converges almost surely to zero in finite time [14]. The sleep-mode models considered in this paper are peculiar in that they reduce to periodic deterministic processes. We will show this without being entirely rigorous. Indeed, as the number of flows approaches infinity, under certain regularity conditions we can apply a law of large numbers argument, which says that

$$X_m^L(t) \to \lambda, \text{a.s., for } L \to \infty.$$

Thus, the amount of traffic arriving at the system during a slot approaches λ, almost surely. For mechanism S_1, we thus find a periodic behavior in which each mobile station goes to sleep for the duration of t_1 slots, collect an amount of λt_1 of traffic, which subsequently decreases again; see Figure 1. The sleep periods of the different MSs do not necessarily coincide, and depend on the initial conditions of the system. This already highlights a problem we will meet later on, namely that the system does not 'forget' completely its past state. We omit the proof that the limiting process is necessarily periodic. Note that for this mechanism the sleep periods have in the limit a fixed length while the lengths of the working periods depend on the initial condition. We therefore explicitly keep track of the offsets with which the MSs go into sleep mode in the limiting process in the random variables $D_m^{(1)} \in \{0, \cdots, t_1-1\}$. More precisely, $D_m^{(1)}$ denotes the last time that MS m goes from active mode to sleep mode, relative to the last time MS 1 went into sleep mode. Obviously, $D_1^{(1)} = 0$. Note that this choice keeps the D_m time-invariant for the limiting process $\mathbf{Q}_m^\infty()$.

For the mechanism S_2, the situation is slightly different in that the sleep periods in the limit do not have a fixed length but the combined sleep and working periods are fixed to the cycle length or a multiple thereof (see Figure 2). Also for this mechanism we need to keep track of the periodic cycle starts, which we summarize into the variables $D_m^{(2)} \in \{0, \cdots, L-1\}$.

3.3 Outline

In this section, we formulate assumptions under which a large-deviations principle for the arrival processes can be transformed by means of the contraction principle [10] into a large-deviations principle of the queue content process.

The contraction principle pushes through LDPs from one topological space to another, under the condition that the mapping between the spaces is continuous. The usual programme (as elucidated for example in [10]) for deriving an

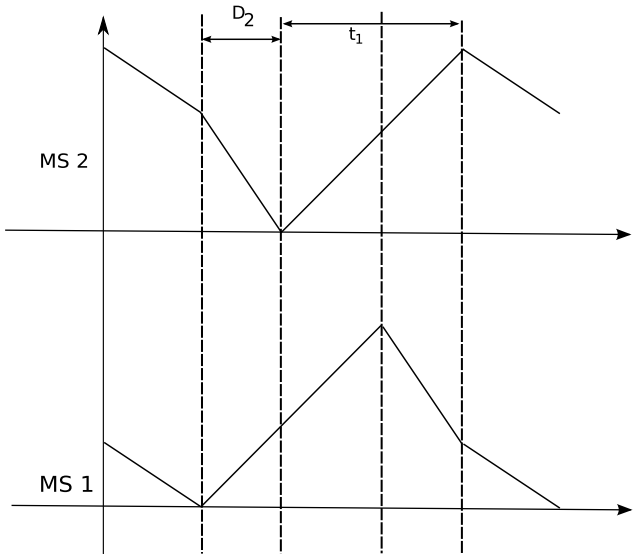

Figure 1: The limiting process $\mathbf{Q}_m^\infty$ for the mechanism S_1. The parameter t_1 and the random variable D_2 are illustrated. Remark that as $\mathbf{Q}_m^\infty$ is a discrete-time process, the straight lines should in fact be replaced by discrete dots.

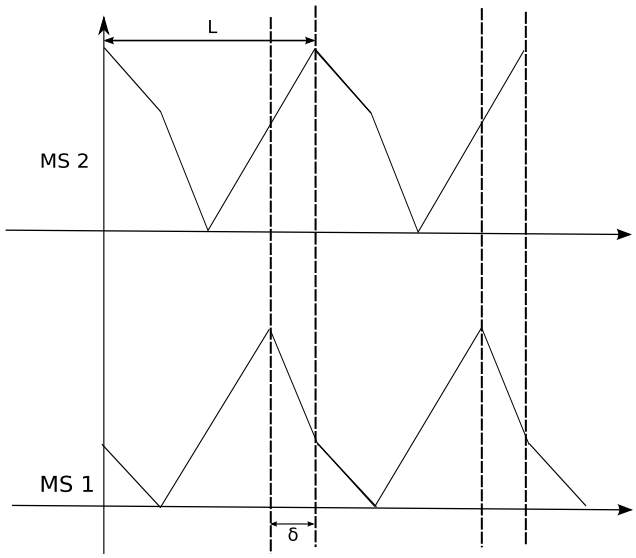

Figure 2: The limiting process $\mathbf{Q}_m^\infty$ for the mechanism S_2. The parameters L and δ are illustrated. Also here, as $\mathbf{Q}_m^\infty$ is a discrete-time process, the straight lines should in fact be replaced by discrete dots.

LDP for a new situation is thus as follows:

1. Take the LDP for the input processes (arrival processes).
2. Define a mapping between the inputs and the output of interest. Examples of the latter are buffer content at time 0, the buffer content sample path is a possible output as well, but a bit harder to obtain.
3. Tweak the topology and/or the mapping such that the mapping is indeed continuous. Recall that a mapping $f : \mathcal{X} \to \mathcal{Y}$ is continuous if the inverse image of every open set of $\mathcal{Y}$ is an open set in $\mathcal{X}$. A possible choice is the so-called initial topology [9].
4. Strengthen the LDP such that it holds for the topology found in the previous point. We often observe a trade-off: the stronger the topology for $\mathcal{X}$, the more mappings that are continuous, but the harder it is to strengthen the LDP. In essence, we must establish exponential tightness of the sequences of measures under the topology.
5. Application of the contraction principle gives us the desired LDP in the topological space $\mathcal{Y}$.

For the LDP results presented here, we need the following assumption on the scaled arrival processes $\mathbf{X}_m^L$.

CONDITION 1. **(Finite-time characteristics)** *For $\boldsymbol{\theta} \in \mathbb{R}^t$, define*

$$\mathbf{\Lambda}_t^L(\boldsymbol{\theta}) = \frac{1}{L} \log \mathbb{E} \exp(L\boldsymbol{\theta} \cdot \mathbf{X}^L(0,t]).$$

Assume that for each t and $\boldsymbol{\theta}$, the limiting cumulant generating function

$$\mathbf{\Lambda}_t(\boldsymbol{\theta}) = \lim_{L\to\infty} \mathbf{\Lambda}_t^L(\boldsymbol{\theta})$$

exists as an extended real number, and that the origin belongs to the interior of the effective domain of $\mathbf{\Lambda}_t$, and that $\mathbf{\Lambda}_t$ is an essentially smooth, lower-semicontinuous function.

Due to the Gärtner-Ellis Theorem [9], the finite-time truncations over $[0,t)$ satisfy an LDP with good rate function

$$\mathbf{I}(\mathbf{x}) = \sup_{\boldsymbol{\theta}} \boldsymbol{\theta} \cdot \mathbf{x} - \mathbf{\Lambda}_t(\boldsymbol{\theta}). \tag{5}$$

Some useful further assumptions include (which simplify matters for example in the numerical computations of Section 4):

1. The arrival process has (asymptotically) independent increments, that is, the rate function $\mathbf{I}(\mathbf{x})$ can be written as:
$$\mathbf{I}(\mathbf{x}) = \sum_k I(x_k), \tag{6}$$
for a certain good rate function $I : \mathbb{R}^M \to \mathbb{R}^+$. Examples include (but are not limited to) the case that the original arrival process is indeed time-independent and identically distributed.
2. Asymptotically 'Markov' increments: there exists a space $\mathcal{Z}$ and good rate functions $K_0(.), K(.|z), I(.|z) : \mathcal{Z} \to \mathbb{R}^+$, such that
$$\mathbf{I}(\mathbf{x}) = \inf\{K_0(y_0) + \sum_j I(x_j|z_j) + \sum_j K(z_j|z_{j-1}) | z_j \in \mathcal{Z}\}. \tag{7}$$

This kind of structure arises for example when taking the many-flows limit for many Markovian arrival flows, see eg. [12].

3.4 Main Result

Let X be the set of arrival processes to the M buffers over a finite interval $[0,T]$, let Y_0 be the set representing the possible initial states (buffer content plus time spent in sleep mode), and let Y be the set of the 'output process': the combined buffer content and energy consumption evolutions in interval $[0,T]$. The function $q : Y_0 \times X \to Y$ maps the input process together with the initial condition to to the output process. Assume that Y_0 and X satisfy LDPs over the Euclidean topology with good rate functions $J_0(.)$ and $I(.)$ respectively.

THEOREM 1. *An LDP holds for the queue content and energy consumption processes over finite time interval* $[0,T]$ *and finite buffer content* b*, with good rate function* $J : Y \to \mathbb{R}^+$*,*

$$J(y) = \inf\{J_0(y_0) + I(x) | y = q(y_0, x), x \in X, y_0 \in Y_0\} \quad (8)$$

PROOF. Let τ_1 be the initial topology over $Y_0 \times X$ with respect to the mapping q, that is, the weakest topology that makes the function $q(.,.)$ continuous. We first establish an LDP over $Y_0 \times X$ under this topology, and then the desired LDP over Y follows by the contraction principle. In contrast to similar proofs, the initial topology τ_1 is neither weaker nor stronger than the Euclidean topology. Hence we must take a detour via their intersection topology τ_I. This is the topology whose set of open sets is the intersection of the open sets of the other two topologies. To push the LDP from the Euclidean topology τ_e down the coarser τ_I, an application of the contraction principle with the identity function suffices. To push it back up to the stronger topology τ_1 with the help of the inverse contraction principle takes a little more work. The identity function is again the mapping of choice, but now we must also show that the sequence (Y_0^L, X^L) is exponentially tight. This is automatically fulfilled if the space $(Y_0 \times X, \tau_1)$ is compact. While this is not the case (consider a sequence with the queue content approaching zero while the MS stays active), we instead compactify the space and prove that the rate function is infinite in the added points. As $(Y_0 \times X, \tau_1)$ is a semimetric topology, we may verify compactness by showing that any sequence in $Y_0 \times X$ has a convergent subsequence. We do so inductively: prove that we can find a subsequence that is convergent for the first time instant. If $T = 1$ then the proof is complete, otherwise we have reduced the problem to a shorter interval. □

4. NUMERICAL METHOD

Relatively few publications on large deviations (especially in telecommunications) have a focus on the numerical solution of the LDP as a variational problem. In the domain of statistical physics, such numerical methods have been investigated among others in [11]. This is partly understandable as it runs counter to the often cited promise of large-deviations to provide solutions where other methods fall short, especially numerical methods suffering from the curse of dimensionality. In this work however, we have implemented a few algorithms that solve a discretized version of the variational problem (8), partly in order to check the optimal solutions that were obtained by purely analytical means, and partly in order to extend the range of scenarios that we can tackle. The results we find are generally promising.

In order to stratify the discussion we focus on a specific large deviation event, namely the case that one specific MS overflows (i.e. reaches an overflow level b). Also, we restrict our attention to arrival processes satisfying either (6) or (7). In the former case, we start out by discretizing the state space $\mathcal{X}$ into a discrete set $\tilde{\mathcal{X}}$ with the help of a discretizing function $\delta : \mathcal{X} \to \tilde{\mathcal{X}}$. It is to be expected (but hard to prove rigourously) that a sufficiently fine grid leads to a solution that is close to the solution of the original problem. The variational problem then reduces to the problem of finding the shortest path in a graph with an edge from node v to node v' having a cost $\inf\{I(x) : v' = \delta(q(v,x))\}$.

In case of (7), we discretize the space $\mathcal{Z}$ of the arrival process as well leading to a discretized space $\tilde{\mathcal{X}} \times \tilde{\mathcal{Z}}$.

This can be solved by means of various existing shortest-path algorithms, of which Dijkstra's algorithm is perhaps the most famous. The time complexity of this class of algorithms is typically $O(E + V \log V)$, while space complexity is $O(V)$, where E is the number of edges and V is the number of vertices. Keeping the number of edges reasonably small is key to a reasonably low computation time. In the numerical examples, we restricted the rate function I of the arrival process to a finite region D when the state space got too large (e.g. for scenarios with multiple mobile stations):

$$\hat{I}(x) = \begin{cases} I(x), & \text{if } x \in D \\ \infty, & \text{otherwise.} \end{cases} \quad (9)$$

We can also replace the standard Dijkstra algorithm by the so-called A^* algorithm or a variant thereof. This algorithm can exploit heuristic bounds on the path length. Especially in structured problems such as the one at hand, huge gains can be made.

Note that even if we resort to numerical techniques to solve the variational problem (8), its computational cost is cheaper than the more straightforward method of extracting information from a Markov chain: direct computation of the stationary vector has (in absence of further structural properties) a $O(V^3)$ time complexity.

5. PRACTICAL APPLICATION

The LDP as described in Theorem 1 involves a variational problem that is in general hard to solve. In this section, we look at a number of specific situations. For reasons of analytical tractability, we limit ourselves to time-independent (uncorrelated) arrival processes. Of course, the variational formulation also holds for a broader class of both short-range and long-range dependent arrival processes, but in those cases analytical expressions are more scarce.

We attempt to find the optimal (i.e. least unlikely) path for the buffer associated with MS 1 to reach a certain level b. The state of the system at time instant 0 is given by the

buffer contents $\mathbf{Q}_m(0), 1 \leq m \leq M$.

There are a couple of heuristic rules to which an optimal path often adheres. Of course, the buffer of MS 1 overflows because there temporarily is a higher amount of traffic than can be served. But by the presence of multiple MSs and by sleep mode, this principal event can be strengthened or weakened. For example, in case of mechanism S_1, it may be 'cheaper' to have an interval in which no arrivals occur first, such that a longer sleep interval will be initiated during which the buffer content can increase faster. Secondly, there is the role of the other MSs. If there are many MSs active, the service rate as seen by MS 1 is lower, so it may be beneficial to have more MSs active than average during the overflowing path. A last feature that will help us is that because of the time-independence of the arrival processes, optimal paths are usually piecewise linear (linear geodesics). We will illustrate the various scenarios by making use of the Brownian motion arrival process, as this typically leads to nice closed form solutions. The reason for this is that the rate function has a quadratic form:

$$I(x) = \frac{(x-\lambda)^2}{2V}. \tag{10}$$

[I must add some more info on the discrete-time Brownian motion arrival process.]

5.1 Single Mobile Station under S_1

We take a look at the relatively simple system with one mobile station. First, we look at mechanism S_1. We are given the buffer content $Q(0)$ at time instant 0 and $S(0)$, the amount of slots that MS 1 is in sleep mode at time instant 0, and T, the interval in which the overflow to level b must take place. Let us first assume that $Q(0) = 0$ and $S(0) = 0$. It is intuitively clear that an optimal path must have a form as shown in Figure 3. It consists of three linear segments, a first of length $\tau_{i-1} := \sum_{k=1}^{i-1} t_k$ during which the arrival rate is zero. Secondly, a segment of length t_i during which the MS resides in the ith sleep interval and the arrival rate is given by the unknown x, and lastly a segment where the MS is active and during which the buffer content b is reached, and which has an unknown length t.

The rate function associated with this path is equal to:

$$I_q(b) = \inf_{i,x,y} \tau_{i-1} I(0) + t_i I(x) + t I(y). \tag{11}$$

From the figure we see the following relation between the different variables:

$$b = x t_i + (y - C)t,$$

so that we can eliminate t. It is noteworthy that the optimal slopes x^* and y^* generally do not depend on the optimal value i^*. Indeed, as the rate function I is differentiable, we can find the optimum by differentiating the above expression to x and y. We find:

$$\begin{cases} t_i \left(I'(x) - \frac{I(y)}{y-C} \right) = 0 \\ \frac{b - x t_i}{(y-C)^2} ((y-c) I'(y) - I(y)) = 0. \end{cases} \tag{12}$$

from each of which t_i vanishes. For the special case of Brownian motion, with mean λ and variance V, we find explicitly the following simple expressions for x^* and y^*:

$$x^* = y^* = 2C - \lambda.$$

It is interesting to note that the optimal arrival rate y^* in the last segment is the same as predicted in a normal single server queue. Our analysis also shows that when overflow level b is very large, the last segment will dominate and we get the same tail behavior as in a system without sleep mode. This agrees with our intuition about large buffer asymptotics. Of course, in wireless systems with lots of delay-intolerant data, we expect buffers to be rather small. The optimization problem is thus reduced to finding the minimum among the i, which is a finite number for most practical situations.

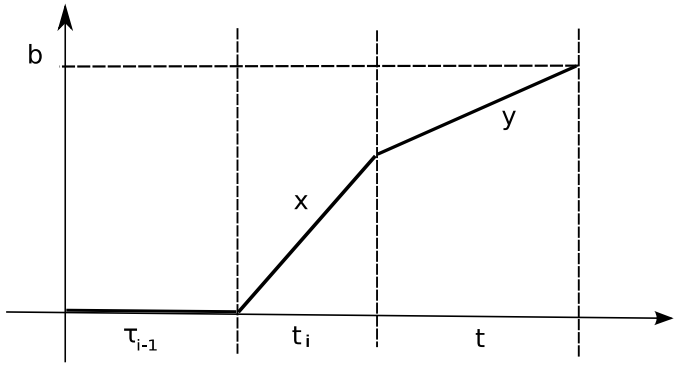

Figure 3: The optimal path for a single MS situation under mechanism S_1.

5.2 Single Mobile Station under S_2

Next, we investigate the situation in which a single mobile station operates under mechanism S_2. We assume that the buffer is empty at the beginning of the excursion and the current sleep cycle is projected to end during slot L_0. There are two candidate optimal paths, both shown in Figure 4. For path A, there are no arrivals during the first L_0 slots, so that the buffer can fill during an entire sleep cycle. For path B, the buffer immediately starts to fill. It is intuitively clear that when L_0 is small, path A will be better, and when L_0 is close to L, then path B will be better. The rate functions of the paths are respectively

$$I_A(b) = \inf_{x,y} L_0 I(0) + L I(x) + t I(y),$$

where $b = xL + (y - C)t$, and

$$I_B(b) = \inf_{x,y} L_0 I(x) + t I(y),$$

where $b = xL_0 + (y - C)t$. The optimal arrival rates are again independent of b, L and L_0, and take the same values for Brownian motion as in the previous scenario. Having found the optimal values x^* and y^*, we easily derive that path A is the optimal path if

$$\frac{L_0}{L} \leq \frac{x^* I(y^*) - (y^* - C) I(x^*)}{(y^* - C)(I(0) - I(x^*)) + x^* I(y^*)},$$

and otherwise path B is better.

5.3 Scenarios with Two Mobile Stations (under mechanism S_2)

Now we move on the situation with 2 MS competing for the same bandwidth. As we will see, the path to overflow is considerably more complex than in the single MS case.

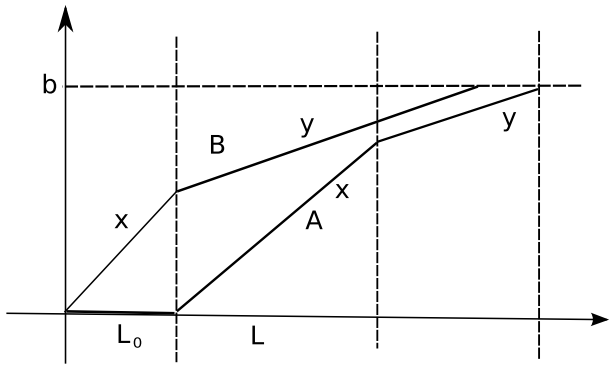

Figure 4: The optimal paths for a single MS situation under mechanism S_2.

We especially pay attention to the influence of the offset δ between the two sleep cycles. The excursion starts with two empty buffers. We discern again a buildup during a sleep period and then a second phase in which the MS actually reaches level b. The difference is that there is now a second MS that can help make the overflow more probable by being more active than average. The unknowns x_1 and x_2 denote the arrival rates during the different phases of MS 1, and y_1, y_2 denote the arrival rates for MS 2. Unknown s denotes the fraction of the time that the MS 2 is in sleep mode during the last phase (which has length t). As MS 1 is continuously active in this last phase, MS 2 sees a service rate $C/2$, and hence it will be active for a fraction $1-s$ if the total traffic during this period is a fraction $(1-s)$ of the amount of traffic that the system can consume during this period (namely $\beta+\frac{1}{2}Ct$). Because of the linear geodesic property, this rate must be constant. Hence we have

$$I_q(b) = \delta I(y_1)+(L-\delta)I(y_2)I((1-s)(\beta+\frac{1}{2}Ct)/t)+LI(x_1)+tI(x_2). \tag{13}$$

We can derive two further relationships between the unknowns: $b = x_1 L + t(x_2 - C + \frac{1}{2}sC)$ and $\beta = \delta y_1 + (y_2 - C)(L-\delta)$.

It is possible to find closed-form expressions for the case of Brownian motion also in this case, but the expressions get dauntingly large. We can however see the impact of the offset δ: if the two cycles would be synchronized, then the 'expensive' segment corresponding to rate y_2 (it is expensive because the BS is serving at full rate C) vanishes, which results in more likely overflows.

5.4 Energy consumption

What does the large-deviations analysis tell us about the other factor in the trade-off, namely the energy consumption ? Firstly, there is the (quite crude) law-of-large-numbers result, that says that the average energy consumption for a large number of sources will be as observed in the limiting paths.

Secondly, information can also be drawn from the large deviations on the average energy consumption over a long period of time. This measures how likely it is to deviate from the energy consumption as predicted by the limit results.

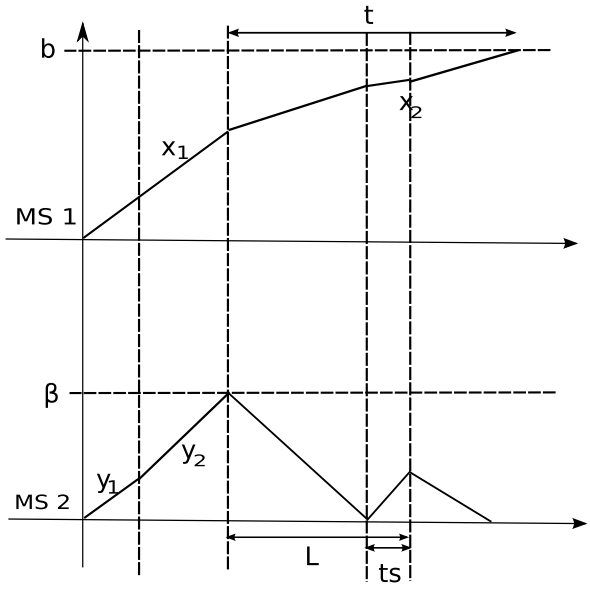

Figure 5: The optimal path for a scenario with 2 MS under mechanism S_2.

5.5 Numerical illustrations

In this section, we show some of the results we obtained by numerical means. The first example we look at considers a 1 MS scenario, with $t_i = i$ for $i \leq C$ and $t_i = C + (i-C)L$ for $i > C$. This boils down to having a closedown period of length C followed by a possibly infinite series sleep periods of the same length L. We assume Brownian arrival processes with $\lambda = 0.7$ and $V = 1.0$. The plots are in accordance with the analytically computed optimal paths in the last section. In case of a non-empty starting condition, we observe a behavior that is quite different from systems without energy-saving mechanism: instead of approaching the overflow level directly, a cheaper path is chosen that visits the boundary and thus profits from the sleep mode mechanism. Of course, if the initial level is very close to the final level, then the direct path may be optimal.

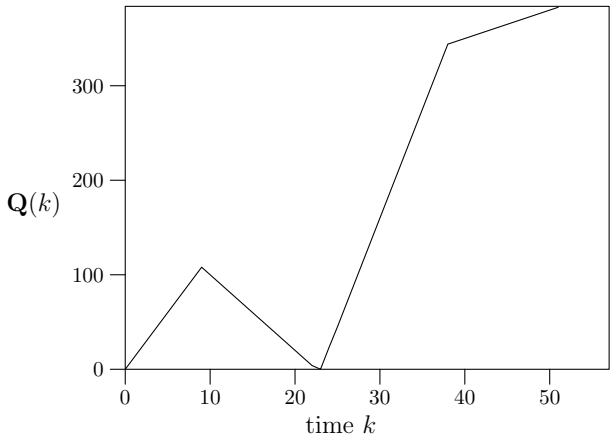

Figure 6: The optimal path for a scenario with 1 MS under mechanism S_1. S_1.

In Figure 7, we plot the decay rate I associated with optimal paths to level $b = 128$ against the length L of the final sleep periods, for different closedown period lengths C. We observe three different regimes. Firstly, for small L, the closedown period is too large for optimal overflow paths to go through a sleep mode phase and hence the direct path is better, resulting in a decay rate that is independent of the C and L. Next, there is a region in which is it profitable to go into sleep mode, but overflow is reached after the sleep period. For the final regime, the length L is sufficiently large for the system to reach overflow during one sleep period of length L, which cause the curves to flatten out for large L.

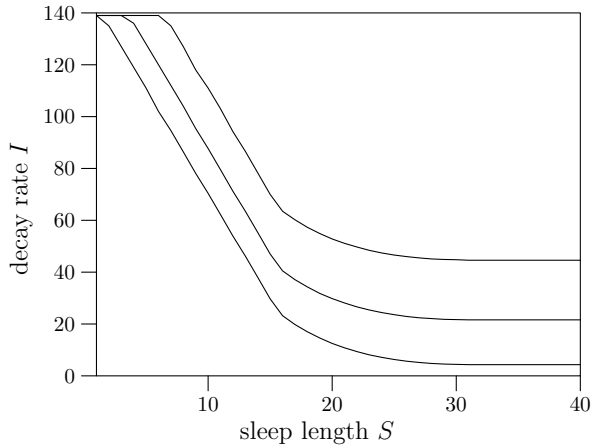

Figure 7: The decay rate I of reaching level $b = 128$ against L for $C = 2, 8, 16$.

Next, in Fig. 8 we look at the most probable overflow path for a 2 MS scenario with sleep mechanism S_1 and bandwidth redistribution. We observe an interesting synchronization phenomenon: before travelling to overflow, the sleep periods of the two MSs first synchronize their work and sleep cycles, so that we have a longer period of shared bandwidth, which makes overflow of one buffer more likely. This synchronization phase is quite messy to analyze in closed form.

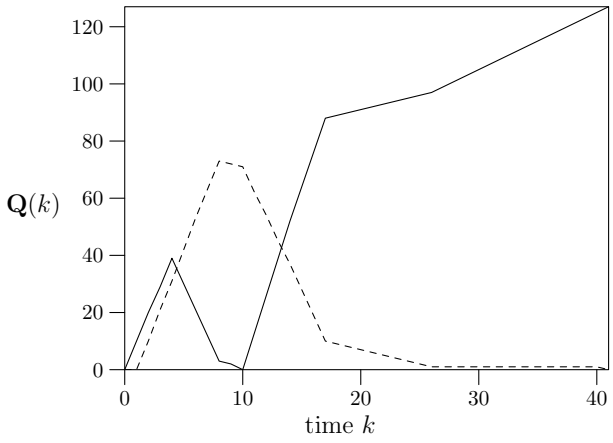

Figure 8: An optimal path for the two MS case with sleep mechanism S_2.

This synchronization phenomenon is much less effective in the 2 MS system with mechanism S_2 (at least when the offset parameters δ_m are chosen in a dispersed manner, for example 0 and $L/2$).

6. GENERAL OBSERVATIONS AND CONCLUSIONS

The paths to overflow get progressively more complex as the number of MS increases. There are however a number of interesting observations which we can extract from the variational formulation, even though we either cannot solve it explicitly or the explicit solution is too complex to be useful.

In the previous subsection, we already touched upon the fact that the offset with which the sleep modes of the different MS arise are a very important factor. We see significantly worse performance when the different MS have synchronized sleep and active periods, both in terms of energy consumption and in terms of overflow probability. This is an area where mechanism S_2 has a marked advantage compared with mechanism S_1, as in the former the offsets are fixed and can be chosen by the base station so as to guarantee a good spread. For mechanism S_1, the sleep periods might drift and end up more or less synchronized (although this drift is a rare event under the many-sources limit), thus making the overflow more likely. There is another feature that can be exploited under S_1 but not under S_2, namely the fact that we can have a long stretch without arrivals, so that the MS enters a long sleep interval, during which the buffer content can rise to high levels. This effect plays an even larger role for bursty traffic.

Although we cannot yet draw definitive conclusions, it appears that from a many-sources point of view, mechanism S_2 seems preferable over the other.

7. REFERENCES

[1] IEEE 802.16e-2005, "Part 16: Air interface for fixed and mobile broadband wireless access systems — Amendment 2: physical and medium access control layers for combined fixed and mobile operation in licensed bands — Corrigendum 1," February 2006.

[2] The Draft IEEE 802.16m System Description Document, IEEE 802.16m-08/003r4, July 2008.

[3] Sleep mode operation for IEEE802.16m, C802.16m-08/721r1, July 2008.

[4] Keep-awake mechanism for 802.16m sleep mode, C802.16m-08/718, July 2008.

[5] K. De Turck, S. De Vuyst, D. Fiems, and S. Wittevrongel, "Performance analysis of the IEEE 802.16e sleep mode for correlated downlink traffic," *Telecommunication Systems,* Vol. 39, No 2, pp. 145–156.

[6] K. De Turck, S. De Vuyst, D. Fiems, S. Wittevrongel and H. Bruneel *Performance of the Sleep-Mode Mechanism of the New IEEE 802.16m Proposal for Correlated Downlink Traffic* Proceedings of NetCOOP 2009, Eindhoven, Oct. 2009

[7] N.-H. Lee, S. Bahk, *MAC sleep mode control considering downlink traffic pattern and mobility*, Proceedings of the IEEE 61st Vehicular Technology Conference, VTC2005-Spring (Stockholm, 30 May–1 June 2005), Vol. 3, pp. 2076–2080.

[8] A. A. V. Anatolii A. Puhalskii. A large deviation principle for join the shortest queue. *Mathematics of Operations Research*, 32(3):700–710, august 2007.

[9] A. Dembo and O. Zeitouni. *Large deviations techniques and applications*, volume 38 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, second edition, 1998.

[10] A. Ganesh, N. O'Connell, and D. Wischik. *Big Queues*. Springer, 2004.

[11] C. Giardinà, J. Kurchan, and L. Peliti. Direct evaluation of large-deviation functions. *Phys. Rev. Lett.*, 96(12):120603, Mar 2006.

[12] M. Mandjes and A. Ridder. A large deviations analysis of the transient of a queue with many Markov fluid inputs: approximations and fast simulation. *ACM Trans. Model. Comput. Simul.*, 12:1–26, January 2002.

[13] N. O'Connell. Large deviations for queue lengths at a multi-buffered resource. *Journal of Applied Probability*, 35(1):240–245, 1998.

[14] P. Robert. *Stochastic Networks and Queues*. Stochastic modelling and Applied Probability. Springer verlag, Berlin, 2003.

[15] D. Wischik. The output of a switch, or, effective bandwidths for networks. *Queueing Systems*, 32:383–396, 1999.

[16] D. J. Wischik. Sample path large deviations for queues with many inputs. *The Annals of Applied Probability*, 11(2):379–404, 2001.

[17] P.E. Hart, N.J. Nilsson, B. Raphael. B. (1968) A formal basis for the heuristic determination of minimum cost paths *IEEE Transactions on Systems Sciences and Cybernetics*, SSC-4(2): 100-107, 1968.