Equivalent Bandwidth Adaptation with Energy Preservation under Delay Constraints

Roberto Bruschi, Franco Davoli
Department of Electrical, Electronics and Telecommunications Engineering, and Naval Architecture
National Inter-University Consortium for Telecommunications
University of Genova
Genova, Italy
roberto.bruschi, franco.davoli@cnit.it

Maurizio Mongelli
Institute of Electronics, Computer and Telecommunication Engineering
National Research Council of Italy
Genova, Italy
maurizio.mongelli@ieiit.cnr.it

***Abstract*—The paper investigates the balance between Quality of Service (QoS) and power consumption for a traffic queue. Automatic adaptation to the power of the packet processing engine is sought through a heuristics and an optimal control strategy. The study allows to track the behavior of the system over time, thus avoiding the optimization of the steady state behavior of the system, which is hardly applicable over small time horizons and with high-variable traffics. The algorithms are validated over a very large set of traffic scenarios, buffer size and delay constraints.**

I. Introduction

A. State of the art

Telecommunications operators are today among the major energy consumers of their nations, and they are still increasing their power requirements [1]. This situation has aroused by the almost complete absence of proportionality between the workload and the energy consumption of current commercial network equipment [2].

In virtue of the wide margin between network capacity and real utilization, the following topics should be investigated: **1)** the configuration speed of power management, which is usually very close to incoming packet rates [3]; **2)** the unpredictability of Internet traffic dynamics, especially at fine-grained scales (e.g., packet inter-arrival times) [4]; **3)** the need for simple and computationally lightweight models and optimization algorithms [5].

Recent approaches [6] to power scaling address the problem by optimizing the steady-state behavior of network devices, under the a-priori knowledge of the statistical features in play. Given the long-range dependency of Internet traffic, the approach guarantees some sub-optimality. A continuous adaptation to the system is needed to achieve better performance.

B. Motivation and contribution

Analytical models of traffic queues as those developed in [6], which include power-related quantities of interest, can be exploited in an optimization process, which finds the suboptimal balance between energy consumption and transmission speed. Suboptimality derives from: the certainty equivalent assumption on the traffic sources (the model needs a match between the actual input process and the model parameters), the periodical adaptation of the model parameters and the on-line re-optimization of the control variables (i.e., the model outputs). The present paper follows a different approach, in which a control paradigm is sought to adaptively minimize the energy consumption under QoS contraints, only on the basis of traffic samples, acquired during the system evolution. The paper extends the results presented in [7], by comparing the simulative approach and the application on a real device for optimal control and by formulating and validating some alternatives of the heuristics.

II. The Problem

A. The queue model

We consider here the case of a single traffic queue, which receives a stochastic input rate process $\alpha(t)$, for which some QoS constraints (packet loss, delay, jitter) should be guaranteed. The problem of finding the minimum service rate of the queue θ_n to guarantee the QoS belongs to the well known equivalent bandwidth (EqB) framework [8], for which several solutions exist in the literature (see, e.g., [9]). If, on the other hand, the service rate θ is linked to the power P supplied to the queue processor, corresponding to a given frequency F of the packet processing engine, and F becomes the optimization variable, the problem consists of finding the minimum F to guarantee the QoS. In other words, we are seeking the minimum working frequency of the internal clock of the processing engine capable to sustain the minimum service rate that maintains the QoS under the present traffic conditions. Solutions to this problem are still unknown.
The size of the time intervals in which the frequency is kept constant is denoted by T. The average delay of the queue d^* is the QoS metric of interest.

For the sake of simplicity, we consider a situation in which θ has been already set to the correct EqB threshold θ_n (under the current traffic conditions)[1] and this corresponds to a nominal admissible frequency F_n. Since θ_n is true in steady state (together with the achievement of the QoS), what we ask here is if there exists, in a small portion of time T, a traffic

[1]Since EqB closed-forms are hardly applicable in general conditions to the average delay metric, we derive here a heuristic approximation of the EqB by means of simulation analysis.

978-1-4799-3088-3/14/$31.00 ©2014 IEEE

sample for which the queue is not congested and the frequency may be reduced to a lower value F_{min}, corresponding to a lower service rate θ_{min}. In order to compensate congestion due to excessive frequency reductions, we also consider situations in which, if congestion is under way, the frequency is switched to F_{Max}, with corresponding service rate θ_{Max}. The θ_{min} level is set to $\theta_{min} = 0.7 \cdot \theta_n$ and the same proportional change is applied to $\theta_{Max} = 1.3 \cdot \theta_n$. Frequency switching operations are performed in a such a way that the effect in the long run is not perceived by the steady state QoS and, at the same time, some power consumption is avoided.

A queueing theorist may ask if the problem is well posed or not, because if the queue is set to its EqB working point θ_n, on average, the periods in which $F = F_{min}$ should be compensated by periods in which $F = F_{Max}$ and the global effect should be an energy consumption corresponding to no frequency changes at all (F always set to F_n). The most important outcome of the paper is to show that this is true in some cases only; in other cases, an intelligent frequency adaptation may result in significant energy saving, without any detrimental effect on the QoS. Finding out which system configurations (traffic parameters and QoS levels) belong to one case or the other (under the applied control algorithms) is the final goal to pursue.

B. Problem formulation

Mathematically speaking, the problem consists of finding the optimal sequence of service rate reallocations $\theta^*(\kappa)$ [2], $\kappa = 1, 2, ..., N_\kappa$, $\theta^*(\kappa) \in \{\theta_{min}, \theta_n, \theta_{Max}\}$, in order to minimize $J = \sum_{\kappa=1}^{N_\kappa} \theta(\kappa)$, under the constraints $d(\kappa) \leq d^*, \forall \kappa$, $d(\kappa)$ being the averaged delay per packet, measured in the interval $[\kappa - 1, \kappa]$. In words, this means trying to always use the lowest possible frequency leading to the lowest rate allocation for $\theta(\kappa)$, without affecting the delay constraint d^*. In principle, a solution could be found by evaluating both the cost and the constraint for each possible combination of the sequence of $\theta(\kappa)$, under any possible combination of traffic parameters and buffer size. Since the procedure is computationally intractable, a feedback law is sought for $\theta^*(\kappa)$. This is the aim of the optimal control formulated later on.

III. Heuristics

Before addressing directly the optimal control solution, a simpler heuristics can be stated. It is based on simple measurements, made on the actual buffer occupancy. The heuristics is shown in Fig. 1. Starting from the working state $\theta = \theta_n$, at the beginning of every time interval T_H, the buffer occupancy is sampled and, if it corresponds to a delay d which is lower than the given threshold, $d < a \cdot d^*$, $a \leq 1$, the rate is reduced from $\theta = \theta_n$ to $\theta = \theta_{min}$. The opposite ($\theta = \theta_{Max}$ from $\theta = \theta_n$) happens if $d > b \cdot d^*$, $b > 1$.

The delay d is related to the measured buffer occupancy (or instantaneous queue length) q as $d = q/\theta$, $\theta \in [\theta_{min}, \theta_n, \theta_{Max}]$. Another possibility is to exploit $d = d(k)$ in place of the instantaneous queue length; in this case, the heuristics is driven by the averaged delay over $[k-1, k]$. The two approaches may differ with increasing T_H since looking at the history of the system over $[k-1, k]$ averages the impact on the delay d rather than reacting on the basis of the instantaneous queue length. On the other hand, instantaneous reactions to queue length may be more effective as they continuously track the temporal behavior of the system. The two approaches are tested in the performance evaluation section.

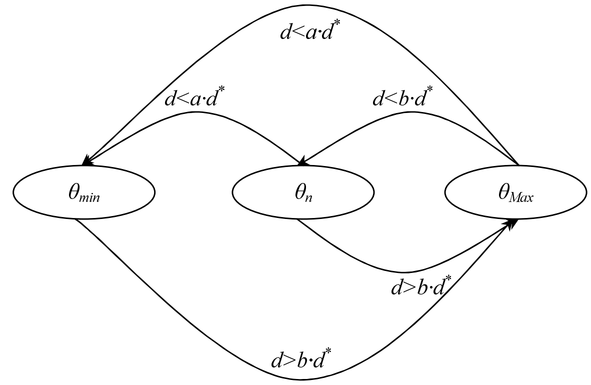

Fig. 1: State Machine of the heuristics.

The state machine of Fig. 1 shows all the possible transitions, e.g., a disruptive congestion condition $d > b \cdot d^*$ under $\theta = \theta_{min}$ may lead to the immediate transition to $\theta = \theta_{Max}$. Parameters a and b are set to 1 and 1.5, respectively. This means the frequency is reduced to the minimum as soon as some delay reduction arises below d^* and the frequency is set to the maximum if the current delay is above 50% of the required QoS. This setting is for sure aggressive in trying to reduce the frequency, but dangerous in terms of QoS. Many other settings may be provided, e.g., more conservative on parameter a with respect to the QoS, but, as a result, with less energy preservation. Even more sophisticated updates may be investigated, e.g., by looking at d in some time periods before the update by using exponential time averaging, but, as we deal in all cases with heuristics for which theoretical guarantees may be hardly found, we fix here only the setting outlined above. Surprisingly, the heuristics can provide much more energy preservation than expected.

IV. Optimal Control

A. Information vector

A sequence of reallocation steps $k = 1, 2, ...,$ is defined such that, on the basis of feedback acquired during the system evolution, a control law $f(\cdot)$ decides on the reallocation at each discrete time instant $t_k = kT$ with $\theta(k) = f(I(k))$. The quantity $I(k)$ is a finite-dimensional information vector collecting observations of some features of interest acquired during the system evolution in the time interval $[k-1, k]$. The components of $I(k)$ must concern quantities correlated with the performance, such as indications about the statistical properties of $\alpha(t)$ (e.g., number of active sources, burstiness) or, more simply (and often less effectively), its mean and variance. $I(k)$ assumes the following form:

$$\boldsymbol{I}(k) = [\ \theta(k), d(k), d^*(k), N(k), Bp(k), \overline{\tau}(k), \overline{\phi}(k), B(k), B_{Max}(k)\ ] \quad (1)$$

[2] In virtue of the mapping of $\{F_{min}, F_n, F_{Max}\}$ into $\{\theta_{min}, \theta_n, \theta_{Max}\}$, in the rest of the paper, we will refer to the θ variables to point out frequency (F) changes with the corresponding rate (θ) allocations.

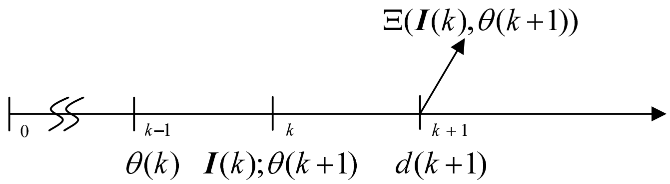

Fig. 2: Evaluation of cost function $\Xi(\cdot)$.

where $\theta(k)$ is the queue rate assigned during the $[k-1,k]$ horizon, $\theta(k) \in \{\theta_{min}, \theta_n, \theta_{Max}\}$, N is the number of active traffic sources giving origin to α, Bp, $\overline{\tau}$ and $\overline{\phi}$ are the peak bandwidth, the average burst size and the average silence duration of the sources, respectively, B and B_{Max} are the current and maximum buffer sizes, respectively.

B. Receding Horizon

We formulate a control approach under the *receding horizon* paradigm, in which the control law acting at time k is optimized by looking at the immediate effect on the subsequent time horizon ($[k, k+1]$) after the reallocation and by disregarding the future evolution of the performance (i.e., after time $k+1$) [10], [11] [3]. The control law is driven by a new cost function, $\Xi(\cdot)$, different from J, but capable to capture both the satisfaction of the delay constraint and the minimization of the rate allocation. In other words, the cost function $\Xi(\cdot)$ is useful to understand in which conditions a frequency reduction is admissible for the time interval $[k, k+1]$, without violating the delay constraint.

Fig. 2 above summarizes one step of the building process of $\Xi(\cdot)$. At time k, the information vector is collected ($\boldsymbol{I}(k)$), a control action is performed ($\theta(k+1)$) and, at time $k+1$, $\Xi(\cdot)$ is computed with respect to the measured delay ($d(k+1)$). Several steps are repeated with respect to other samples of $\boldsymbol{I}(k)$ and $\theta(k+1)$. A regression scheme is then applied to the corresponding samples of $\Xi(\cdot)$ in order to infer $\Xi(\cdot)$ for every admissible $\boldsymbol{I}$ and θ (the details of the regression scheme are provided in the following). Since $\Xi(\cdot)$ is defined to achieve a global minimum in correspondence of the minimum θ satisfying d^* (over $[k, k+1]$), the control law at time k consists of minimizing $\Xi(\boldsymbol{I}(k), \theta)$ with respect to θ. The $\Xi(\cdot)$ function recalls the Q_1-function of *Reinforcement Learning* (RL) [12], which corresponds to a 1-step optimization over the dynamical system chain.

C. Collection of data

The collection of the data driving the $\Xi(\cdot)$ building process may derive directly from the real system because it depends only on samples acquired during the system evolution. We use a simulator here as it is simpler than interacting with the real system. The approach can be however easily extended to both sampling and control on the real system as explained later. Differently from sampling the real system, the simulator also allows the evaluation of $d(k+1)$ with respect to $\boldsymbol{I}(k)$ under both $\theta(k+1) = \theta_{min}$ and $\theta(k+1) = \theta_n$. The minimum of $\Xi(\boldsymbol{I}(k), \theta)$ is thus set in correspondence of θ_{min}, θ_n or θ_{Max}, depending on the achieved $d(k+1)$ (i.e., if it satisfies d^* or not). This can be easily done by replicating the same simulation scenario twice after k (see Fig. 2 again) and by driving the two replications with θ_{min} and θ_n, respectively. The θ_{Max} option is not tested because it defines anyway the minimum of $\Xi(\cdot)$ if $\theta(k+1) = \theta_n$ is not sufficient to obtain $d(k+1) \leq d^*$. In more detail, the simulation steps are as follows. **1)** A realization of the process $\alpha(t)$ is generated under some traffic parameters, extracted from a random distribution (the same is done for setting the buffer size at the beginning of the simulation), **2)** after a steady state has been reached, $\theta(k)$ is randomly chosen in $\{\theta_{min}, \theta_n, \theta_{Max}\}$ in order to collect in the information vector any admissible rate allocation belonging to the previous time interval $[k-1, k]$, **3)**, after T, the information vector $I(k)$ is collected, $\theta(k+1)$ is assigned by evaluating the two alternatives $\theta(k+1) = \theta_{min}$ and $\theta(k+1) = \theta_n$ and **4)** a sample of the cost $\Xi(\cdot)$ is obtained after T in dependence of $\theta(k+1)$ (Fig. 2) under the following rules:

if $(\theta(k+1) = \theta_{min}) \wedge (d(k+1) \leq d^*)$ **then**

$\{\Xi(\boldsymbol{I}(k), \theta_{min}) = -1; \Xi(\boldsymbol{I}(k), \theta_n) = 0; \Xi(\boldsymbol{I}(k), \theta_{Max}) = 1\}$

else if $(\theta(k+1) = \theta_n) \wedge (d(k+1) \leq d^*)$ **then**

$\{\Xi(\boldsymbol{I}(k), \theta_{min}) = 0, \Xi(\boldsymbol{I}(k), \theta_n) = -1, \Xi(\boldsymbol{I}(k), \theta_{Max}) = 1\}$

else if $(d(k+1) > d^*)$ **then**

$\{\Xi(\boldsymbol{I}(k), \theta_{min}) = 1, \Xi(\boldsymbol{I}(k), \theta_n) = 0, \Xi(\boldsymbol{I}(k), \theta_{Max}) = -1\}$

D. Sampling the real system

In the real system, we simply generate a long trajectory of the system under usual operating conditions, except from changing $\theta(k)$, every k, at random or under some heuristics. Several time intervals $[k, k+1]$ are extracted from the trajectory, with the corresponding variables $\boldsymbol{I}(k), \theta(k+1)$ and $d(k+1)$. We can then slightly modify the rules above for setting the minimum of $\Xi(\boldsymbol{I}(k), \theta)$ in dependence of the difference between $d(k+1)$ and d^*. For example, it is intuitive that, if $\theta(k+1) = \theta_n$ and $d(k+1) = 0.5 \cdot d^*$, the minimum of $\Xi(\boldsymbol{I}, \theta)$ should be set in correspondence of $\theta = \theta_{min}$ because the achieved delay (under that specific $\boldsymbol{I}$) was half the expected delay. The advantage of sampling with the simulator relies on validation of delay satisfaction (synthesized into $\Xi(\cdot)$) under measurements through all admissible values of θ with respect to the same sample of $\boldsymbol{I}$, whereas, in the real system, a single sample of the delay satisfaction is derived from a single sample of $\boldsymbol{I}$. In this perspective, the neural scheme inferring the delay satisfaction in correspondence of all admissible $\boldsymbol{I}$ (explained later on) receives more reliable information from the simulator rather than from the real system (in correspondence of the same number of samples).

E. Neural approximation

The control law is then built as follows:

$$\hat{f}(\boldsymbol{I}(k)) = \hat{F}(\min_{\theta} \ \hat{\Xi}(\boldsymbol{I}(k), \theta, \boldsymbol{\omega}^*)) \qquad (2)$$

$\hat{\Xi}(\cdot, \boldsymbol{\omega}^*)$ being an approximation of the function $\Xi(\cdot)$, which is derived from a regular neural regression scheme with optimal

[3] More specifically, since receding horizon control means solving an optimization problem over T' seconds and implementing the control only over the first $\Delta T'$ seconds, our approach consists of 1-step receding horizon, as both performance sampling and control application are relative to the same size of the observation horizon T.

neural weights assignment $\boldsymbol{\omega}^*$, applied to a database containing N_Ξ couples of $\{^\varsigma\boldsymbol{I};\Xi(^\varsigma\boldsymbol{I},\theta)\}$ for each sample ς, ς=1,...,N_Ξ (no matter how the samples are obtained, i.e., simulatively or by sampling the real system). A neural network with all tangent neural units and a single output is used to derive $\hat{\Xi}(\cdot,\boldsymbol{\omega}^*)$, plus a fuzzy rule, $\hat{F}(\cdot)$, which maps the output of the neural network to $\{\theta_{min},\theta_n,\theta_{Max}\}$, depending on the proximity of the neural output to $\{-1,0,1\}$, respectively. The mentioned regression scheme consists of finding the weights assignment $\boldsymbol{\omega}^*$ so that:

$$\sum_{\varsigma=1}^{N_\Xi}[\Xi(^\varsigma\boldsymbol{I},\theta)-\hat{\Xi}(^\varsigma\boldsymbol{I},\theta,\boldsymbol{\omega}^*)]^2\leq\rho. \quad (3)$$

Problem (3) consists of tuning the output of the neural network in order to approximate the collected values of $\Xi(^\varsigma\boldsymbol{I},\theta)$ as a function of the information vectors $^\varsigma\boldsymbol{I}$, ς=1,...,N_Ξ. Typical values for the bound ρ are in the range [0.001, 0.5]. The functional approximation of $\Xi(\boldsymbol{I},\theta)$ with $\hat{\Xi}(\boldsymbol{I},\theta,\boldsymbol{\omega}^*)$ allows to obtain a syntectic representation of $\Xi(\boldsymbol{I},\theta)$ for all $\boldsymbol{I}$ belonging to the space defined by the the samples $^\varsigma\boldsymbol{I}$, ς=1,...,N_Ξ.

V. Performance Evaluation and Discussion

A. Traffic parameters and equivalent bandwidth

On-off traffic is considered; each source is an on-off process with time durations exponentially distributed on ($\overline{\tau}$) and off ($\overline{\phi}$). Both $\overline{\tau}$ and $\overline{\phi}$ have been set to 1.0 s, whereas all the other parameters have been considered as free variables. The packet size amounts to 80 bytes. Variable packet sizes can be included in the analysis without significant changes to the presented algorithms. The other parameters are set as follows: peak bandwidth $Bp\in[15,30]$ kbps, maximum number of connections $N_{Max}\in[5,20]$, buffer size $B_{Max}\in[100,200]$, measured in packets of 80 bytes. The d^* target is a function of θ_n, which is set to $(1.1\cdot\theta_{ref}\cdot N)$, $\theta_{ref}=\frac{B_p}{b}$, $b=\frac{\overline{\tau}+\overline{\phi}}{\overline{\tau}}=2$, θ_{ref} being a heuristic EqB of a single connection and b the traffic burstiness.

B. Performance metrics

The mentioned system is simulated using an ad-hoc C++ simulator implementing the control algorithms to set the frequency of the packet processing engine and measuring the achieved delay over time. An energy cost, J_e, is defined as follows:

$$J_e=\frac{D_{F_n}}{D}+2\cdot\frac{D_{F_{Max}}}{D}-2\cdot\frac{D_{F_{min}}}{D} \quad (4)$$

D being the overall duration of the simulation and $D_{F_{min}},D_{F_n},D_{F_{Max}}$ the overall durations of the periods in which the frequency has been set to F_{min},F_n,F_{Max}, respectively. Negative values of J_e mean more energy preservation; J_e close to 1 means $D_{F_{Max}}\to 0$ and $D_{F_{min}}\to 0$, i.e., the buffer is working on an EqB regime for the largest part of time, J_e close to $\frac{1}{3}$ means a balance among the above durations, i.e., $D_{F_{min}}\cong D_{F_n}\cong D_{F_{Max}}\cong\frac{1}{3}$. A similar delay cost is defined as:

$$J_d=\bar{d}-d^*\ \textbf{if}\ (\bar{d}>d^*),\ 0\ \textbf{otherwise} \quad (5)$$

in order to highlight how far we may be from QoS preservation, $\bar{d}$ being the measured average delay along the entire simulation period.

The neural control of subsection IV-E is applied. The information vector is composed by the quantities outlined in (1), apart from $\overline{\tau}$ and $\overline{\phi}$, which have been kept fixed as said before.

Training. In virtue of the independence of the simulations operated to build the database of the samples $\{^\varsigma\boldsymbol{I};\Xi(^\varsigma\boldsymbol{I},\theta)\}$ outlined in subsection IV-E, the processing time for training can be significantly reduced by using multiple CPUs. The database is built in less than 2 hours over an IntelCore i7@2.4GHz (with 4Gb RAM, 7 CPUs and using Visual Studio C++ and Windows7 64-bit), over which 6 simulation threads may run in parallel. Overall, the database contains 9000 samples of $\{^\varsigma\boldsymbol{I};\Xi(^\varsigma\boldsymbol{I},\theta)\}$. A neural network with 10 hidden neural units is used for the regression scheme (3). The inherent computational time was around 15 minutes.
Test. After training, 300 independent repetitions are generated by extracting the system parameters from a set of random uniform distributions in the ranges outlined in subsection V-A. Each repetition lasts 100s.
Both in training and test, the reference period T for sampling the information vector, changing the frequency accordingly and evaluating the cost function $\Xi(\boldsymbol{I},\theta)$ is 1s.

C. Performance

Together with the neural network (NN), we first consider a comparison between the two alternatives for setting d in the heuristics as outlined in section III, namely, by following either the instantaneous queue length (called q-heu) or the average delay over T_H (called $d(k)$-heu). The a and b parameters of the heuristics are fixed to 1 and 1.2, respectively. We take as a reference the J_d and J_e metrics, averaged over all the repetitions of the test phase; we denote them as $\bar{J}_d$ and $\bar{J}_e$, respectively. The results are shown in table I. The following considerations must be noted.

	$\bar{J}_d$ [ms]
neural network	5
$d(k)$-heu T_H=100ms	24
$d(k)$-heu T_H=1s	38
q-heu T_H=100ms	25
q-heu T_H=1s	33
	$\bar{J}_e$
neural network	0.095
$d(k)$-heu T_H=100ms	-0.086
$d(k)$-heu T_H=1s	-0.007
q-heu T_H=100ms	-0.122
q-heu T_H=1s	-0.126

TABLE I: $\bar{J}_d$ and $\bar{J}_e$ of the test phase.

• $\bar{J}_e$ metric. Surprisingly, the NN has an average energy consumption which is higher than the one of the heuristics. In order to discover energy saving with the NN, a deeper analysis is needed with respect to each single replication of the test phase. The analysis is reported in the following. The lowest energy consumption of the heuristics arises under the q-heu strategy, independently of T_H. This means instantaneous reactions are more effective, in the energy consumption viewpoint, rather than following the average delay, which seems to be too polarized on the history of the observation horizon.
• $\bar{J}_d$ metric. Under the same T_H, the average difference to the delay constraint is invariant between the q-heu and $d(k)$-heu

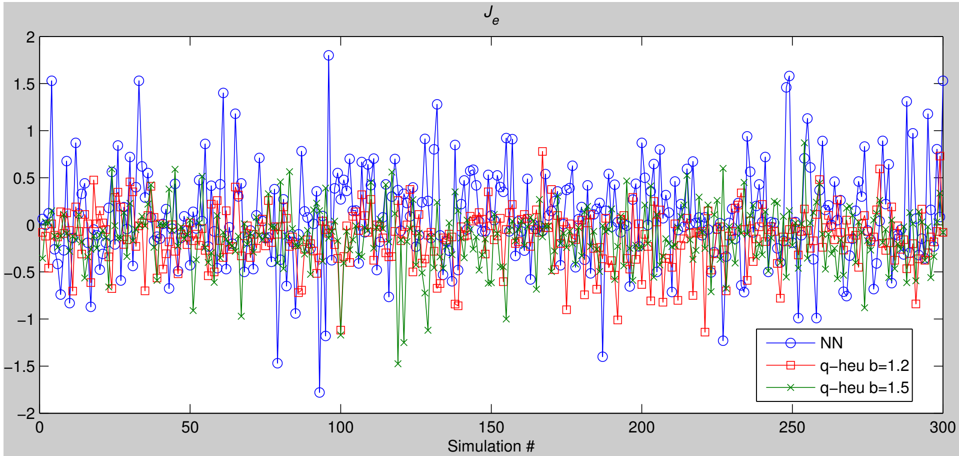

Fig. 3: Energy cost J_e.

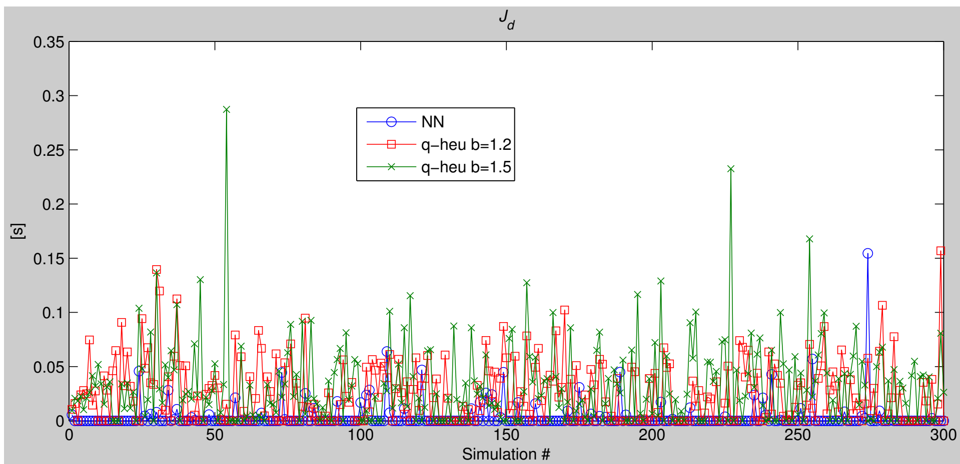

Fig. 4: Delay cost J_d.

approaches. Such a difference is of one order of magnitude higher than with the NN. This is not surprising since the NN is built under the Ξ function, which does not allow power reduction if the delay constraint is not satisifed. The Ξ function can even increase the power allocation if the delay is not satisfied, in specific conditions, under the nominal power level. In this respect, the NN would be preferred to the heuristics as it does not degrade QoS while it searches for a lower energy consumption.

To discover the impact of the system' parameters, we now look at the performance over all the replications of the test phase. A specific setting of the parameters (d^*, Bp, B_{Max} and N_{Max}) corresponds to each independent replication. Fig. 3 and Fig. 4 show the single J_e and J_d values achieved. The shown heuristics follows the q-heu approach under b set to 1.2 and 1.5 and with T_H=100ms.

The heuristics preserves energy in some cases (low peaks of the heuristics in Fig. 3), but with some QoS degradation, as expected from the results of table I (J_d values of Fig. 4 are often much higher than with using the NN). The average J_d of the heuristics in Fig. 4 is 25ms and 30ms with $b = 1.2, 1.5$, respectively. We verified through extensive simulations that a specific parameters setting (a^* and b^*) can be often derived for the heuristics to satisfy the QoS with less energy than the one corresponding to the EqB regime. However, finding a synthetic representation of this mapping (i.e., between a^* and b^* and the system parameters) may be a hard task. On the other hand, the NN guarantees the QoS (as said above, $\bar{J}_d$=5ms in Fig. 4 is 5ms) and obtains an energy saving in several situations (lowest peaks of NN in Fig. 3). In other situations (corresponding to the highest NN peaks of Fig. 3), the NN experiences an energy waste. In order to discriminate these opposite situations, we depict in Figs. 5-6 which system parameters are mapped to J_e values higher than 0.5 (i.e., cases with a significant energy waste) and which others are mapped to $J_e < -0.5$ (i.e., with a significant energy saving). By simply looking at the pictures, very simple rules may be inferred. These rules define whether the NN should replace the EqB allocation (that does not provide any energy saving). Condition $d^* > 150$ms (dashed lines in Figs. 5-7) satisfies $Pr\{J_e < -0.5\} = 78\%$. If also the $N_{Max} > 8$ rule (solid vertical line of Fig. 7) is used in logical 'AND' with the previous one, 90% of $Pr\{J_e < -0.5\}$ is captured.

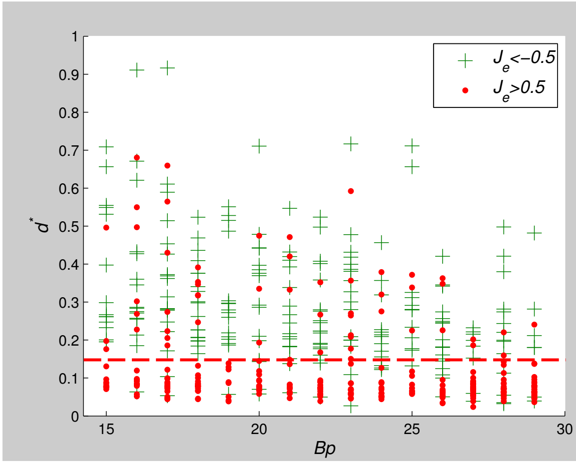

Fig. 5: Bp ([kbps]) versus d^* ([s]).

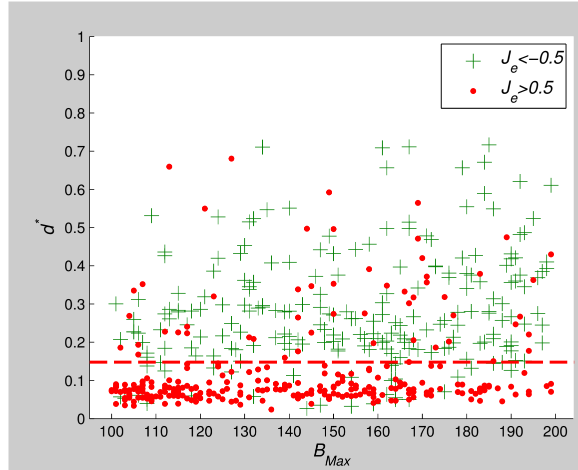

Fig. 6: B_{Max} ([packets]) versus d^* ([s]).

D. Discussion

The above rules hold under the system setting of subsection V-A; in more complicated settings (i.e., larger ranges of the parameters, including the burstiness), more intricate rules may be needed. This topic is currently under investigation, by applying supervised classification tools. It is interesting to note also that no rules arise on the buffer size as the distribution of the points in Fig. 6 is uniformly spread over the admissible B_{Max} range. The sporadic presence of some $J_e > 0.5$ points in the regions of Figs. 5-6 characterized by d^* >150ms leaves the question open whether more powerful techniques may be capable to capture more energy saving. It must be noted however that they could be much more computationally expensive (e.g., using Model Predictive Control), while the proposed approach is computationally light after training. A RL approach could deserve further study in the direction of insightfully exploring QoS and energy metrics, while still being applicable in real time.

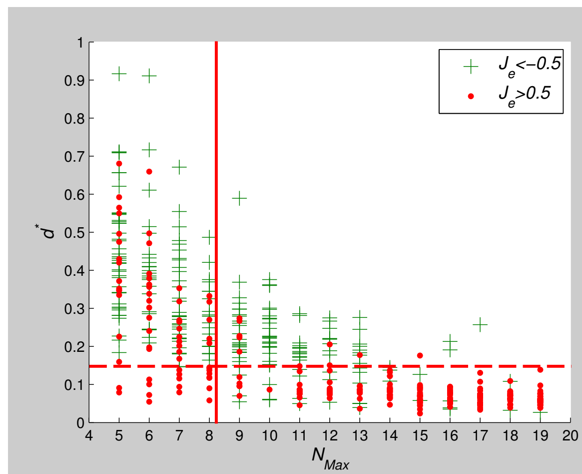

Fig. 7: N_{Max} versus d^* ([s]).

VI. Conclusions and Future Work

The paper presented a heuristics and an optimal control approach to the power control of a traffic queue. The performance evaluation shows how the optimal control guarantees a significant energy saving under a specific set of system conditions. Future work deals with extensions in terms of QoS metrics and with the presence of multiple queues.

References

[1] R. Bolla, R. Bruschi, F. Davoli, F. Cucchietti, "Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures ," IEEE Communications Surveys and Tutorials, vol 13 no. 2, pp. 223-244, May 2011.

[2] S. Nedevschi, L. Popa, G. Iannacone, S. Ratnasamy, D. Wetherall, "Reducing Network Energy Consumption via Sleeping and Rate-Adaptation," Proc. USENIX NSDI'08, San Francisco, CA, USA, April 2008.

[3] S. Henzler, Power Management of Digital Circuits in Deep Sub-Micron CMOS Technologies, Series in Adv. Microel., Springer, New York, 2007.

[4] J.L. Garcia-Dorado, A. Finamore, M. Mellia, M. Meo, M. Munafo', "Characterization of ISP Traffic: Trends, User Habits, and Access Technology Impact," IEEE Trans. Network and Service Management, vol. 9, no. 2, pp. 142-155, June 2012.

[5] B. Alcott, "Jevons' paradox," Ecological Economics, vol. 54, no. 1, pp. 9-21, July 2005.

[6] R. Bolla, R. Bruschi, A. Carrega, F. Davoli, "Green Networking With Packet Processing Engines: Modeling and Optimization," IEEE/ACM Trans. on Network., in press; doi: 10.1109/TNET.2013.2242485.

[7] R. Bruschi, F. Davoli, M. Mongelli, "Adaptive Frequency Control of Packet Processing Engines in Core Networks," IEEE Commun. Letters, submitted.

[8] F. Kelly, "Notes on effective bandwidths," in F. P. Kelly, S. Zachary, I. Ziedins, Eds., Stochastic Networks: Theory and Applications , Royal Statistical Society Lecture Notes Series, vol. 4, Oxford University Press, 1996, pp. 141-168.

[9] L. Aspirot, P. Belzarena, P. Bermolen, A. Ferragut, G. Perera, M. Simon, "Quality of Service Parameters and Link Operating Point Estimation based on Effective Bandwidths," Perf. Eval., vol. 59, no. 23, pp. 103-120, Feb. 2005.

[10] E. Camacho and C. Bordons, Model Predictive Control, Springer-Verlag, New York, 2005.

[11] M. Baglietto, F. Davoli, M. Marchese, M. Mongelli, "Neural Approximation of Open-loop-Feedback Rate Control in Satellite Networks," IEEE Trans. Neural Networks, vol. 16, no 5., pp. 1195-1211, Sept. 2005.

[12] D. Ernst, P. Geurts, L. Wehenkel, L. Littman, "Tree-based Batch Mode Reinforcement Learning," Journal of Machine Learning Research, vol. 6, pp. 503556, Apr. 2005.