A Stochastic Knapsack Model for Energy Efficient Management of Multi-Server Queues

Raffaele Bolla[†‡], Roberto Bruschi[†‡], Alessandro Carrega[†‡], Franco Davoli[†‡], Chiara Lombardo[†‡], Beatrice Siccardi[†‡]
[†] Department of Electrical, Electronic and Telecommunications Engineering, and Naval Architecture (DITEN), University of Genoa, Italy
[‡] National Laboratory of Smart and Secure Networks (S2N) of the National Inter-university Consortium for Telecommunications (CNIT), Genoa, Italy
raffaele.bolla@unige.it | roberto.bruschi@unige.it | alessandro.carrega@cnit.it | franco.davoli@unige.it | chiara.lombardo@unige.it | beatrice.siccardi@tnt-lab.unige.it

***Abstract*— We consider multiple bursty flows characterized by different statistical parameters and performance requirements, which generate packets that require some form of processing at the network edge. We model the processing resources as multiple servers that can be activated/deactivated on a longer time scale with respect to the packet dynamics. Packets awaiting service are queued in an infinite buffer, and the queueing model adopted is of the $M^X/D/C$ type; the flow dynamics is instead represented by a birth-death model on a much longer time scale. By exploiting a time-scale decomposition, we describe possible Call Admission Control (CAC) strategies that are based on a Stochastic Knapsack model over the space of flows that satisfy packet-level delay constraints. We compare a Complete Partitioning and a Complete Sharing CAC scheme in terms of energy efficient implementation.**

Keywords— Energy Efficient Networking, Adaptive Rate, MEC, Load Balancing, Energy-Performance Tradeoff.

I. INTRODUCTION

Following a period of increased interest at the beginning of the century, the issue of energy efficiency in networking – particularly in the fixed network portions of access, transport and core, and in networking devices, spawned by some pioneering works on various aspects of management and analysis ([1]-[6] among others; see also [7] and [8] for survey papers on that period) – has seen a lesser momentum over the successive decade. Attention seemed to have shifted more on datacenters [9], [10] and the wireless segment (e.g., [11]). However, it is worth noting that the Key Performance Indicators (KPIs) regarding the fifth generation of mobile wireless networks (5G) did not include specific figures on energy efficiency, though significant reductions in energy consumption were expected. This lack of specification was likely due to the belief that the advent of network softwarization and virtualization technologies would increase energy efficiency per se, owing to consolidation of resources in the presence of low traffic load but actually neglecting the fact that the widespread use of general-purpose hardware may jeopardize energy saving, unless proper control strategies are put in operation [12].

On the other hand, the topic of network energy efficiency has received again increased attention in the evolution toward the 6th generation mobile network (6G), where very ambitious goals are being set also with respect to energy efficiency [13]. In the light of this renewed interest, it is worth focusing attention not only on the wireless segment, where specific technological innovations may suggest novel modeling and control approaches, but also on the access, backhaul and core network, and to aim to end-to-end strategies for energy efficiency that also include the increased presence of in-network and edge computational elements, as those introduced by Mobile Edge Computing (MEC) [14]-[16]; a recent survey on energy efficiency in the "telco cloud" is contained in [17].

In doing so, the powerful modeling and control techniques that were devised for "traditional" networking equipment can be revisited with virtualized architectures and MEC computational resources in mind. Dynamic flow-based models, queueing models suitable for parametric optimization, and machine learning (ML) techniques (see, e.g., [18] for ML-based control approaches that do not neglect the relevance of analytical modeling of dynamic systems) are all tools that are worth considering in this framework.

In this paper, we elaborate further on the modeling and control scheme that we introduced in [19] for the load-adaptive adjustment of processing capacity to serve incoming streaming flows, under specific packet-level delay and flow-level blocking constraints. The goal of adopting the minimum amount of processing resources needed to satisfy performance requirements is pursued to indirectly obtain a reduction in energy consumption. Putting processing units (cores) to sleep – i.e., to low power states – in the presence of low workloads and waking them up when an increase in workload would jeopardize performance, is equivalent to de-activating/activating servers in the queueing model and may significantly reduce the power consumption without negatively affecting performance. Indeed, the relation between power consumption and the number of active cores in multi-core processors has been investigated, among others, in [20], [21], where the consumption is shown to be roughly proportional (or piecewise linear) with respect to the number of active cores at a given operating frequency.

Herein, we rely on an $M^X/D/C$ queuing model [22] to represent the processing units that perform a specific network function on packets that are queued for service; however, we explicitly consider here that packets may be generated by flows

979-8-3503-6595-5/24/$31.00 ©2024 IEEE

that are characterized by different statistical features and performance requirements. Then, based on a Service Separation concept and on Stochastic Knapsack models [23], we devise admission control strategies for the flows.

The paper is organized as follows. We define the model structure in Section II. Numerical results are presented and commented in Section III. Finally, section IV contains the conclusions and directions for further research.

II. MULTI-CLASS MULTI-SCALE TRAFFIC MODELS

We consider streaming data flows (we will use the terms "stream" and "flow" interchangeably in what follows) that need some kind of processing (e.g., User Plane Functions (UPFs) that perform packet recognition and redirection) and are directed to multiple micro-datacenters in the edge. We assume the statistical distribution of flows to be given by a birth-death model: flows are generated by a Poisson distribution with parameter λ_f and have an exponentially distributed duration with parameter μ_f; moreover, they are subjected to an Admission Control, in order to avoid overloading the computational resources. We also assume that accepted flows carry batches of packets with exponentially distributed interarrival times and that the batch length is characterized by a random variable X with discrete long-tail distribution (Zipf) with mean β [packets/batch]; packet processing times are assumed to be of deterministic duration D. The latter assumption is approximately justified in some Use Cases (e.g., UPFs' header processing and lookup table search), and basically shifts the random nature of processing times over packet batches, owing to the random length of the latter.

We model queueing and processing at the packet level as an $\mathrm{M^x/D/C}$ multi-server queueing system, where C represents the number of active cores of our processing units. With system utilization less than 1, such model is known to admit a stationary distribution that can be determined analytically in closed form [22]. However, data streams may be characterized by different statistical parameters, in terms of average flow and batch generation rates. Therefore, at the flow level, we would be in the presence of a Generalized Stochastic Knapsack [23]. As suggested in [23], the most advisable and manageable operational procedure in this case is based on the concept of Service Separation; namely, packets generated by flows with the same statistical characteristics and performance requirements are multiplexed together in a separate queue and are assigned a certain number of specific processing units. In other words, flows with similar characteristics belong to a specific class, and computational resources are assigned on a per-class basis. Under Service Separation, we will focus on the Admission Control strategy known as *Complete Partitioning* (CP), where each class is assigned exclusively up to a maximum number of processing units.

On the other hand, there is also the possibility (though with some limitation in generality) to handle analytically a Stochastic Knapsack without Service Separation, corresponding to mixing all traffic classes in a single queue, and adopting the Admission Control strategy known as *Complete Sharing* (CS). We will also consider this case and use it for comparison with CP in the specific situation in which all traffic steams are characterized by the same processing time for packets.

Then, let us focus on a specific edge datacenter with the availability of an overall processing capacity resource pool of $C_{\max}$ units (e.g., maximum number of cores that can be activated); we consider K different stream classes and let $\lambda^{(k)}$ be the batch generation rate, $\beta^{(k)}$ the average batch length, and $D^{(k)}$ the processing time of class-k packets. Class k flows' arrival rates and average durations are represented by $\lambda_f^{(k)}$ and $1/\mu_f^{(k)}$, respectively.

Our goal is twofold, and it will be pursued at two different levels of granularity in the traffic units: namely, packet-level and flow-level. At the packet level, we want to find the minimum processing capacities $C_{\min}^{(k)} \leq C_{\max},\ k = 1, \dots, K$, that are required to satisfy packet-level Quality of Service (QoS) requirements for each given number $m^{(k)}$ of active (i.e., accepted in the system and generating packet batches) class-k streams. We will express packet-level QoS requirements in terms of a single KPI per class, represented by an upper bound on the average waiting time of the class packets. The allowable combinations of flows of the different classes that satisfy such QoS requirements ("flow profiles") determine the so-called "schedulable region" or "feasibility region" in the space of flows. Then, at the flow level, we will specify the two CS and CP admission control strategies. In the case of CP, the boundaries of the partitions will be obtained through a parametric optimization procedure to minimize a weighted average of the blocking probabilities of the flows.

II.A. $M^x/D/C$ Model and Average Waiting Times

We consider the k-th processor's queue conditional to the number of active class-k flows $m^{(k)}$. Any change in $m^{(k)}$ would produce a variation in the total offered load $m^{(k)}\lambda^{(k)}\beta^{(k)}$ in [pkts/s], and thus a transient behaviour of the queueing system; however, since variations in the flow dynamics are expected to occur on a time scale much longer than that of batch interarrival times and packet service times (which determine the queue dynamics under a given $m^{(k)}$), we can approximate the conditional probability distribution of the number of queued packets with its stationary expression, which would be reached under the condition that

$$\rho^{(k)} = \frac{m^{(k)}\lambda^{(k)}\beta^{(k)}D^{(k)}}{C^{(k)}} < 1 \qquad (1)$$

(we consider here the values $C^{(k)}$ of computational resources – i.e., servers in the queue – assigned to class k to have also been fixed). We note that a more rigorous justification of our stationary approximation, based on Courtois' decomposition, can be found in reference [24] in a related problem involving the relation between guaranteed-bandwidth and best-effort traffic. Condition (1) imposes an upper bound on the maximum number of flows $m^{(k)}$ that can be accepted for a given $C^{(k)}$:

$$m^{(k)} < \left\lceil \frac{C^{(k)}}{\lambda^{(k)}\beta^{(k)}D^{(k)}} \right\rceil - 1 \equiv m_{\max}^{(k)} \quad (2)$$

where $\lceil x \rceil$ represents the smallest integer greater than or equal to x. Then, following [22], the average queue length $L_q^{(k)}\left(m^{(k)}, C^{(k)}\right)$ of the M^x/D/C system with $m^{(k)} \geq 1$ active flows and $C^{(k)}$ servers can be written analytically as

$$L_q^{(k)}\left(m^{(k)}, C^{(k)}\right) = \frac{1}{2C^{(k)}(1-\rho^{(k)})}\left[\left(C^{(k)}\rho^{(k)}\right)^2 + \right.$$
$$-C^{(k)}\left(C^{(k)}-1\right) + \sum_{j=2}^{C^{(k)}-2}\left[C^{(k)}\left(C^{(k)}-1\right) - j(j-1)\right]p_j^{(k)} +$$
$$\left. + C^{(k)}\rho^{(k)}\left(E\left\{X^{(k)^2}\right\}/\beta^{(k)} - 1\right)\right] \quad (3)$$

where $X^{(k)}$ represents class-k batch size and $p_j^{(k)}$, $j = 2, \dots C^{(k)} - 2$ are the probabilities of having j packets in the queueing system (the summation in the r.h.s. disappears for $C^{(k)} < 4$).

By applying Little's Theorem, the average waiting time (conditional to the presence of at least 1 active flow) is

$$W_q^{(k)}\left(m^{(k)}, C^{(k)}\right) = \frac{L_q\left(m^{(k)}, C^{(k)}\right)}{m^{(k)}\lambda^{(k)}\beta^{(k)}} \quad (4)$$

II.B. Stochastic Knapsack – Service Separation with Complete Partitioning (CP)

As regards the traffic at the flow level, the Stochastic Knapsack model representing it is characterized by a vector Markov Chain $\underline{n} = \left[n^{(1)}, n^{(2)}, \dots, n^{(K)}\right]$, where the k-th component is a birth-death process with traffic intensity $A_f^{(k)} = \lambda_f^{(k)}/\mu_f^{(k)}$. Let now $C_1, C_2, \dots, C_K$, with $\sum_{k=1}^K C_i = C_{\max}$, be a partition of the available computational resources, and let $C_{\min}^{(k)}\left(m^{(k)}\right)$ be a function defining the minimum amount of computational resources that would be necessary to maintain class-k packet-level constraints for each allowable value $m^{(k)}$ that the stationary random variable $n^{(k)}$ can assume:

$$C_{\min}^{(k)}\left(m^{(k)}\right) = \min\left\{0 < C^{(k)} \leq C_k : W_q^{(k)}\left(m^{(k)}, C^{(k)}\right) \leq \right.$$
$$\left. \leq \widehat{W}_q^{(k)}, \right\}, k$$
$$= 1, \dots, K \quad (5)$$

where $\widehat{W}_q^{(k)}$ is a desired upper bound on the average waiting delay of class-k packets.

Then, the state space of $\underline{n}$ is defined by

$$S := \left\{\underline{n} \in {\mathbb{N}_0}^K : W_q^{(k)}\left(m^{(k)}, C_{\min}^{(k)}\right) \leq \widehat{W}_q^{(k)}, m^{(k)} \right.$$
$$\left. = 0,1, \dots, \bar{m}_{max}^{(k)}, k = 1, \dots, K\right\} \quad (6)$$

where $\bar{m}_{\max}^{(k)}$ is the maximum number of active class-k flows such that

$$W_q^{(k)}\left(\bar{m}_{\max}^{(k)}, C_k\right) \leq \widehat{W}_q^{(k)} \quad (7)$$

S represents the feasibility region corresponding to a Complete Partitioning admission control policy in the space of flows within which the packet-level constraints are satisfied. At this point, we can still trade-off power consumption (related to the number of active processing units) and per-class blocking probabilities $P_B^{(k)}$.This can be done either by setting given upper bounds $\bar{P}_B^{(k)}, k = 1, \dots, K$, or by minimizing a weighted sum of the per-class blocking probabilities with respect to the partition coefficients. We have chosen the second alternative here. Since the flow dynamics are described by a birth-death Markov Chain with traffic intensity $A_f^{(k)}$ (giving rise to a M/M/$m_{\max}^{(k)}$/$m_{\max}^{(k)}$ queueing system), the blocking probabilities in this case are provided by the Erlang B formula

$$ER\left[A_f^{(k)}, \bar{m}_{\max}^{(k)}\right] = \frac{\left[A_f^{(k)}\right]^{\bar{m}_{\max}^{(k)}}/\bar{m}_{\max}^{(k)}!}{\sum_{j=0}^{\bar{m}_{\max}^{(k)}}\left[A_f^{(k)}\right]^j/j!} \quad (8)$$

Then, we seek the partition $C_1, C_2, \dots, C_K, \sum_{k=1}^K C_i = C_{\max}$ that minimizes a weighted sum of the blocking probabilities; i.e., we want to find

$$\min_{\substack{C_1, C_2, \dots, C_K \\ \sum_{k=1}^K C_i = C_{\max}}} \sum_{k=1}^K \frac{A_f^{(k)}}{A} ER\left[A_f^{(k)}, \bar{m}_{\max}^{(k)}(C_i)\right] \quad (9)$$

having defined $A = \sum_{k=1}^K A_f^{(k)}$. This optimization problem can be solved numerically in different ways. Since the objective function is separable in the optimization variables, Dynamic Programming can be applied, as in reference [23], p. 122; another possibility is to use a descent method, by exploiting the convexity in the number of servers of the Erlang B and of its analytic continuation [25], [26]. In summary, the management and control framework that has been introduced allows: *(i)* to maintain average delay constraints at the packet level with the minimum allowable energy consumption, by activating only (over the time scale of flow dynamics) the necessary number of processing units; *(ii)* to further trade-off performance and power consumption, by means of the choice of the resource partitions that minimize the weighted average (over the classes) of blocking probabilities.

II.C. Stochastic Knapsack – Complete Sharing (CS)

As already mentioned, in the case in which the packets of all traffic streams require the same processing time, we can also configure a CS Admission Control strategy. Indeed, since the batch generation model of active flows is Poisson, the sum of batches generated by active flows would still be a Poisson flow. Of this aggregate *single* flow, where all $C_{\max}$ units are shared, we only need to know the batch arrival intensity, along with the first and second moments of the batch lengths.

The batch arrival intensity averaged over all classes can be derived by applying the total probability theorem to the average batch arrival intensities of each class, by considering that the probability of arrival of a flow of class k is given by $\lambda_f^{(k)}/\lambda_f$, where $\lambda_f = \sum_{j=1}^K \lambda_f^{(j)}$:

$$\bar{\lambda} = \sum_{k=1}^{K} \frac{\lambda_f^{(k)}}{\lambda_f} \lambda^{(k)} \quad (10)$$

Analogously, we can define the average batch length over all classes as

$$\bar{\beta} = \sum_{k=1}^{K} \frac{\lambda_f^{(k)}}{\lambda_f} \beta^{(k)} \quad (11)$$

Moreover, given the independence of the batch lengths of the various classes, if $\sigma_{(k)}^2$ represents the variance of the batch length of class k, we have for the variance σ^2 of the aggregate batches:

$$\sigma^2 = \sum_{k=1}^{K} \frac{\lambda_f^{(k)}}{\lambda_f} \sigma_{(k)}^2 \quad (12)$$

and, consequently, for the mean square value $\overline{X^2}$ of the aggregates

$$\overline{X^2} = \sigma^2 + \bar{\beta}^2 \quad (13)$$

Given the presence of m flows of the aggregate traffic and C active computational resources, we can write the system utilization as

$$\rho = \frac{m\bar{\lambda}\bar{\beta}D}{C} \quad (14)$$

and the condition on m for $\rho < 1$ as

$$m < \left\lfloor \frac{C}{\lambda_f \bar{\beta} D} \right\rfloor - 1 \equiv m_{max} \quad (15)$$

Finally, if the processing times of packets for all classes are the same ($D^{(k)} = D, k = 1, \dots, K$), we can re-write the equivalent of expression (3) for the aggregate traffic (given $m \geq 1$):

$$L_q(m, C) = \frac{1}{2C(1-\rho)} [(C\rho)^2 + \\ -C(C-1) + \sum_{j=2}^{C-2} [C(C-1) - j(j-1)] p_j + \\ + C\rho\left(\overline{X^2}/\bar{\beta} - 1\right)] \quad (16)$$

and, consequently, for the average delay

$$W_q(m, C) = \frac{L(m, C)}{m\bar{\lambda}\bar{\beta}} \quad (17)$$

Now we can define

$$C_{min}(m) = \min\{0 < C \leq C_{max} : W_q(m, C) \leq \bar{W}_q\} \quad (18)$$

where

$$\bar{W}_q = min\left(\hat{W}_q^{(1)}, \dots, \hat{W}_q^{(K)}\right) \quad (19)$$

is the most stringent requirement over the classes. An incoming flow belonging to the aggregate will be accepted if

$$C_{min}(m+1) \leq C_{max} \quad (20)$$

III. Numerical Results

We analyze here various numerical results for the determination of the minimum number of active cores that respects the requirements at both packet and flow level. The results were obtained by iteratively finding the minimum number of active cores which satisfies the packet-level constraint (see Eqs. (4) and (5) in the CP case, and Eqs. (17)-(19) in the CS case) and finding the partition that minimizes the flow-level objective function in the CP case (see Eq. (9)). We consider two classes, whose parameters, unless otherwise stated, are shown in Table I.

TABLE I. Parameter values

Parameter	Numerical value
$C_{\max}$	24
$\lambda^{(1)}$	20 Mbatches/s
$\beta^{(1)}$	2.3 pkts/batch
$D^{(1)}$	10 ns
$A_f^{(1)}$	10
$\hat{W}_q^{(1)}$	50 ns
$E\{X^{(1)^2}\}$	12.5
$\lambda^{(2)}$	3.5 Mbatches/s
$\beta^{(2)}$	1.9 pkts/batch
$D^{(2)}$	100 ns
$A_f^{(2)}$	0.1
$\hat{W}_q^{(2)}$	100 ns
$E\{X^{(2)^2}\}$	6.5

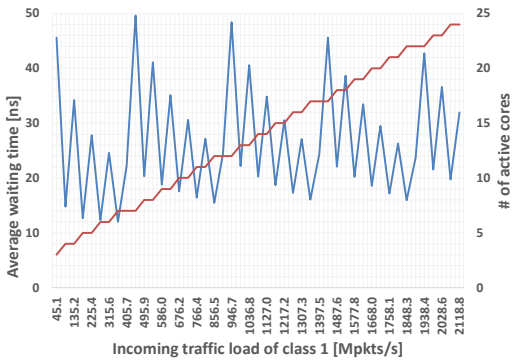

Fig. 1. Average packet waiting time (blue) and minimum number of active cores (red) for the Multi-Flow Multi-Server Queuing Model conditional to having $m^{(1)} = 0,1, \dots, \bar{m}_{max}^{(1)}$ accepted flows for class 1 and with $m^{(2)} = 1$.

We are going to consider first the results conditional to having $m^{(k)}$ accepted flows for class k. Therefore, the incoming traffic load of class k corresponds to:

$$m^{(k)} \lambda^{(k)} \beta^{(k)} \quad (21)$$

with $m^{(k)} = 0,1, \dots, \bar{m}_{max}^{(k)}$.

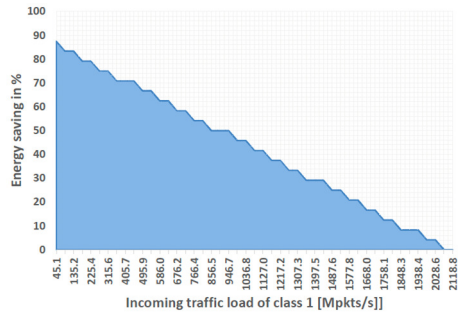

Fig. 2. Energy saving for the Multi-Flow Multi-Server Queuing Model conditional to having $m^{(1)} = 0,1,\dots,\bar{m}_{max}^{(1)}$ accepted flows for class 1 and with $m^{(2)} = 1$.

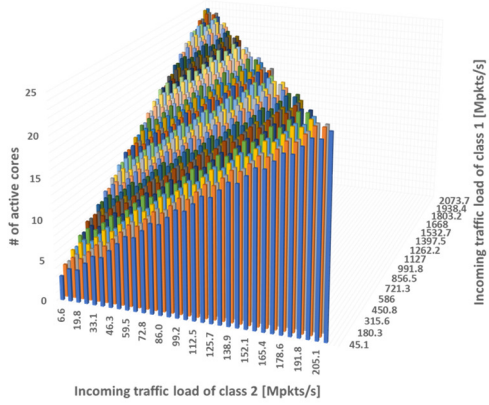

Fig. 3. Minimum number of active cores for the Multi-Flow Multi-Server Queuing Model conditional to having $m^{(1)} = 0,1,\dots,\bar{m}_{max}^{(1)}$ accepted flows for class 1 and $m^{(2)} = 0,1,\dots,\bar{m}_{max}^{(2)}$ accepted flows for class 2.

Results for class 1 are shown in Figs. 1 and 2. Herein, we consider the model having a fixed number of accepted flows of class 2, with $m^{(2)} = 1$. With respect to a single-flow queue the difference lies in the addition of the minimum number of cores for class 2 (in this case equal to 2 in order to satisfy the upper bound on the average waiting time), which is constant. Therefore, it is possible to highlight that the plots are similar to those in our previous work [19]. Fig. 1 shows the average waiting time and the minimum number of activated cores. The former increases until it reaches a depth which corresponds to an additional activated core to respect the constraint on the average waiting time, while the latter has an almost "step-wise" plot. Fig. 2 shows the energy saving in percentage that is gained with the dynamic activation of the cores. Comparing all the plots with [19], we can highlight that here the energy savings are lower: graphically, in Fig. 2 the parts where the plot is parallel to the x-axis are shorter. This can be attributed to the different packet-level constraint: in [19] we considered an upper bound on the waiting time averaged with respect to the number of accepted flows, while here we consider the exact waiting time conditional to the number of active flows. Therefore, herein, the constraint is stricter, as the upper bound is satisfied for every single number of accepted flows. Now, let us introduce the results for the CP Multi-flow Multi-server queue: the number of accepted flows in the range $\{0,1,\dots,\bar{m}_{max}^{(k)}\}, k = 1,2$. Figs. 3 and 4 show the results for the number of active cores and the energy saving respectively. Again, it can be noticed that the plots in both figures have a "step-wise" trend. This trend is noticeable in two directions: one following the incoming traffic loads of class 1 and one following the incoming traffic load of class 2.

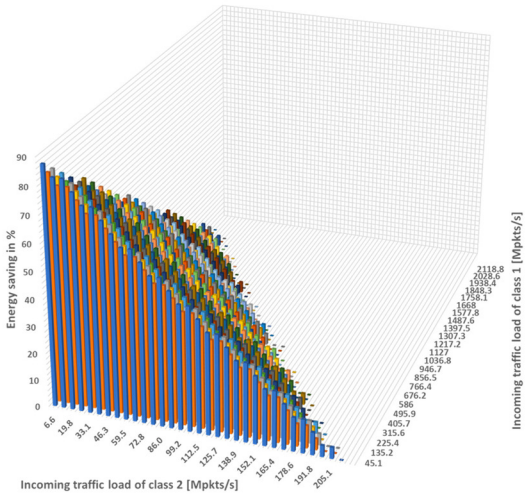

Fig. 4. Energy saving (in percentages) for the Multi-Flow Multi-Server Queuing Model conditional to having $m^{(1)} = 0,1,\dots,\bar{m}_{max}^{(1)}$ accepted flows in class 1 and $m^{(2)} = 0,1,\dots,\bar{m}_{max}^{(2)}$ accepted flows for class 2.

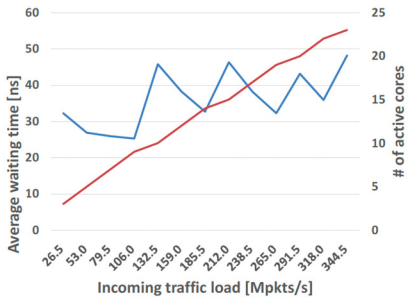

Fig. 5. Average packet waiting time (blue) and minimum number of active cores (red) for the Complete Sharing Queuing Model conditional to having $m^{(2)} = 0,1,\dots,\bar{m}_{max}^{(2)}$ accepted flows.

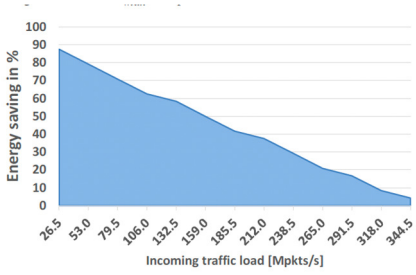

Fig. 6. Energy saving (in percentages) for the Complete Sharing Queuing Model conditional to having $m^{(2)} = 0,1,\dots,\bar{m}_{max}^{(2)}$ accepted flows.

Finally, we compare the CP and CS cases. Following the details in Section II.C, let us lose some generality by setting the service time: $D^{(1)} = D^{(2)} = D = 60\ ns$. Figs. 5 and 6 show the results for the CS case. First, we can notice that the plots (the red

one in Fig. 5, and the one in Fig. 6) lose the "stepwise" trend with respect to Complete Partitioning; this is already a qualitative sign of a lower energy saving. Numerically, we can compare the total energy saving (i.e., the area) between the two cases: we suppose CP with the same parameters as in Table I except for the service time (i.e., $D^{(1)} = D^{(2)} = D = 60$ ns) and CS with two fictitious classes both with the same incoming traffic load (the one shown in Figs. 5 and 6). Results show that the *Complete Partitioning* approach allows to save ~2.5 times the energy with respect to the *Complete Sharing* one.

IV. Conclusions

Recently, the topic of network energy efficiency has received again increased attention in the evolution toward 6G. In the light of this renewed interest, we have presented a modelling and control scheme for the load-adaptive activation of processing capacity to serve incoming flows, while satisfying requirements at packet and flow level. In detail, we have considered an $M^X/D/C$ queuing model that serves flows with different statistical features and performance requirements. Flows are described by a Stochastic Knapsack model over the feasible region defined by packet-level constraints, and subject to CP and CS admission control. Regarding the requirements, at packet level we have considered an upper bound on the average waiting time, while at flow level our aim is to minimize the weighted average of blocking probabilities. The final goal is to activate only the minimum number of cores. Results show that our proposed model can save energy while satisfying the requirements. Furthermore, we have compared the proposed model with one relying on another admission strategy: *Complete Sharing*. Results show that the Complete Partitioning model allows to save approximately 2.5 times more energy.

Acknowledgment

This work has been partially supported by the Horizon 2020 5G-PPP Innovation Action 5G-INDUCE (Grant Agreement no. 101016941) and by the Horizon Europe 6G IA Research and Innovation Action 6Green (Grant Agreement no. 101096925).

References

[1] M. Gupta and S. Singh, "Greening of the Internet," *Proc. ACM SIGCOMM Conf. (SIGCOMM 03)*, Karlshue, Germany, Aug. 2003, pp. 19-26.

[2] K. Christensen, B. Nordman, R. Brown, "Power Management in Networked Devices," *IEEE Computer*, vol. 37, no. 8, pp. 91-93, Aug. 2004.

[3] C. Gunaratne, K. Christensen, S. Suen, B. Nordman, "Reducing the Energy Consumption of Ethernet with an Adaptive Link Rate (ALR)," *IEEE Trans. on Computers*, vol. 57, no. 4, pp. 448-461, Apr. 2008.

[4] S. Nedevschi, L. Popa, G. Iannaccone, D. Wetherall, S. Ratnasamy, "Reducing Network Energy Consumption via Sleeping and Rate-Adaptation*," Proc. 5th USENIX Symp. on Networked Systems Design and Implementation (NSDI '08)*, San Francisco, CA, 2008, pp. 323-336.

[5] M. Allman, K. Christensen, B. Nordman, V. Paxson, "Enabling an Energy-Efficient Future Internet Through Selectively Connected End Systems," *Proc. ACM SIGCOMM HotNets Workshop (HotNets 07)*, Atlanta, GA, Nov. 2007.

[6] R. S. Tucker, R. Parthiban, J. Baliga, K. Hinton, R. W. A. Ayre, W. V. Sorin, "Evolution of WDM Optical IP Networks: A Cost and Energy Perspective", *IEEE J. Lightwave Technol.*, vol. 27, no. 3, pp. 243-252, Feb 2009.

[7] 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 Communication Surveys & Tutorials*, vol. 13, no. 2, pp. 223–244, 2nd Quart. 2011.

[8] A. P. Bianzino, C. Chaudet, D. Rossi, J. L. Rougier, "A Survey of Green Networking Research," *IEEE Communication Surveys & Tutorials*, vol. 14, no. 1, pp. 3-20, 1st Quart. 2012.

[9] T. Mastelic, A. Olesiak, H. Klussen, I. Brandic, J.-M. Pierson, A. V. Vasilakos, "Cloud Computing: Survey on Energy Efficiency," *ACM Computing Surveys*, vol. 47, no. 2, Article 33, pp. 33:1-33:36, Dec. 2014.

[10] M. Dayarathna, Y. Wen, R. Fan, "Data Center Energy Consumption Modeling: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 732-794, 1st Qr. 2016.

[11] M. Ismail, W. Zhuang, E. Serpedin, K. Qaraqe, "A Survey on Green Mobile Networking: From the Perspectives of Network Operators and Mobile Users," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1535-1556, July 2015.

[12] R. Bolla, R. Bruschi, F. Davoli, C. Lombardo, J. F. Pajo, O. R. Sanchez, "The Dark Side of Network Functions Virtualization: A Perspective on the Technological Sustainability," *Proc. IEEE International Conf. on Communications (ICC 2017)*, Paris, France, May 2017.

[13] I. F. Akyildiz, A. Kak, S. Nie, "6G and Beyond: The Future of Wireless Communications Systems," *IEEE Access*, vol. 8, pp. 133995-134030, July 2020.

[14] European Telecommunications Standards Institute, "Multi-Access Edge Computing (MEC); Framework and Reference Architecture," ETSI GS MEC 003 v2.2.1, 2020.

[15] F. Giust *et al*., "MEC Deployments in 4G and Evolution towards 5G," ETSI White Paper, vol. 24, pp. 1–24, 2018.

[16] T. Koketsu Rodrigues, J. Liu, N. Kato, "Offloading Decision for Mobile Multi-Access Edge Computing in a Multi-Tiered 6G Network," *IEEE Trans. Emerg. Topics Comput*., vol. 10, no. 3, pp. 1414-1427, Jul.-Sep. 2022.

[17] E.-V. Depasquale, F. Davoli, H. Rajput, "Dynamics of research into modeling the power consumption of virtual entities used in the telco cloud", *Sensors*, vol. 23, art. 255, pp. 1-69, Jan. 2023.

[18] R. Zoppoli, M. Sanguineti, G. Gnecco, T. Parisini, *Neural Approximations for Optimal Control and Decision*, Springer Nature, Cham, Switzerland, 2020.

[19] R. Bolla, R. Bruschi, A. Carrega, F. Davoli, C. Lombardo, "Trading off Power Consumption and Delay in the execution of network functions by dynamic activation of processing units," *Proc. 2022 1st International Workshop on Network Energy Efficiency in the Softwarization Era*, in conjunction with *IEEE NetSoft 2023*, Milan, Italy, June 2022, pp. 1-6.

[20] I. Takouna, W. Dawoud, C. Meinel, "Accurate Multicore Processor Power Models for Power-Aware Resource Management," *Proc. 2011 9th IEEE International Conference on Dependable, Autonomic and Secure Computing*, Sydney, Australia, Dec. 2011.

[21] V. R. Gomes da Silva, C. Valderrama, P. Manneback, S. Xavier-de-Souza, "Analytical Energy Model Parametrized by Workload, Clock Frequency and Number of Active Cores for Share-Memory High-Performance Computing Applications," *Energies*, vol. 15, art. 1213, Feb. 2022.

[22] H. C. Tijms, *A First Course in Stochastic Models*, Wiley, Chichester, England, 2003.

[23] K. W. Ross, *Multiservice Flow Models for Broadband Telecommunication Networks*, Springer-Verlag New York, Secaucus, NJ, 1995.

[24] S. Ghani, M. Schwartz, "A Decomposition Approximation for the Analysis of Voice/Data Integration," *IEEE Transactions on Communications*, vol. 43, no. 7, pp. 2441-2452, July 1994.

[25] E. J. Messerli, "Proof of a Convexity Property of the Erlang B Formula," *Bell System Technical J*ournal, vol. 51, pp. 951-953, 1972.

[26] A. A. Jagers, E.A. Van Doorn, "On the Continued Erlang Loss Function," *Operations Research Letters*, vol. 5, no. 1, pp. 43-46, June 1986.