# Trading off Power Consumption and Delay in the Execution of Network Functions by Dynamic Activation of Processing Units

Raffaele Bolla*† Roberto Bruschi*† Alessandro Carrega* Franco Davoli*† Chiara Lombardo†

*Department of Electrical, Electronic and Telecommunications Engineering, and Naval Architecture (DITEN)
University of Genoa, Italy
{*name*}.{*surname*}@unige.it

†National Laboratory of Smart and Secure Networks (SN2) of
the National Inter-university Consortium for Telecommunications (CNIT), Genoa, Italy
{*name*}.{*surname*}@cnit.it

***Abstract*—Beside increasing flexibility and programmability, the current network "*softwarization*" trend is believed to be beneficial also in respect of energy efficiency, owing to the consolidation of resources made possible by virtualized networking components. However, the widespread use of general-purpose hardware may jeopardize energy saving, unless proper control strategies are put in operation. In this context, the paper addresses a "*smart sleeping*" control problem, where computing resources in multi-core processors executing network functions are modelled as multi-server queues, and the number of active processing units (either physical or virtual) can be dynamically adjusted by parametric control over a time scale compatible with the long-term dynamics of the traffic flows that require processing. We show that, on average, up to $25\,\%$ of processing capacity of a network node can be turned off in the presence of bursty traffic with low load without significantly affecting packet latency.**

***Index Terms*—Energy efficiency, MEC, hardware offloading**

## I. Introduction

OVER the past two decades, the aspect of energy efficiency of telecommunications equipment and networks has been a concern, spawned by the increasing attention toward a "*green*" approach to the industrial evolution. Starting from data center operations [1] then covering wireless networking [2] and the "*traditional*" Internet Protocol (IP) fixed network [3], the attention on the theme has been slightly decreasing with the advent of network virtualization and "*softwarization*" brought forth by Software Defined Networking (SDN) [4] and Network Functions Virtualization (NFV) [5], with some notable exceptions [6]. Indeed, with the advent of virtualization technologies, wired-wireless integration, Mobile Edge Computing (MEC) and the greater presence of cloud-native applications at the network edge, the attention to energy efficiency aspects has tended to shift more to the wireless segment, in the belief that virtualization would increase energy efficiency of the fixed network, owing to consolidation of resources in the presence of low traffic load.

We believe that the topic is bound to the forefront again with the evolution toward the 6th generation mobile network (6G), where very ambitious goals are being set also with respect to energy efficiency [7]. However, in this context "*softwarization*" and increased network automation per se would not be sufficient to increase energy efficiency, especially in the fixed network segment. Notwithstanding the consolidation of resources made possible by virtualized networking components [8], the widespread use of general-purpose hardware may jeopardize energy saving, unless proper control strategies are put in operation [9].

To trade off energy consumption and performance, suitable dynamic control techniques are needed. Possible approaches to derive them may stem from dynamic flow-based models, from queueing models suitable for parametric optimization, or from machine learning techniques (see, e.g., [10]).

The approach we take in this paper relies on queueing models and on the adaptive adjustment of the processing capacity, on a longer time scale with respect to the dynamics of queues. We adopt an $\mathrm{M}^x/\mathrm{G}/N$ queueing model [11] to represent the processing units that perform a specific network function on packets that are queued for service. According to the model, packets arrive in batches. The latter are supposed to be generated by flows that require the specific treatment offered by the given network function. The dynamics of flows is also represented by a queueing model, with time scales much longer than those characterizing burst arrivals and packet service times.

This overall representation of flows and bursts can be used, in one case, to model the offloading of network functions to specialized programmable hardware to implement Physical Network Functions (PNFs) that substitute Virtual Network Functions (VNFs) in critical applications requiring very fast processing. These may be encountered, among other cases, in the MEC environment, where packets incoming from

This work has been partially supported by the Horizon 2020 5G-PPP Innovation Action 5G-INDUCE – Grant Agreement no. 101016941 – and by the Horizon 2020 Innovation Action SPIDER – Grant Agreement no. 833685.

applications running on User Equipment (UE) devices to MEC attach points may need to be filtered and directed to external networks for further processing, rather than being forwarded to the mobile network backhaul. On the other hand, in the more frequent cases where VNFs are implemented via software in containers, the servers in the queueing system may represent the virtual processors, and the model can be adopted to decide when new processing instances of the containerized VNF should be activated to support horizontal scaling. Obviously, the power consumption of PNFs and VNFs will be different: in the virtualized case, it would refer to the fraction of physical processing units utilized by the virtual entity, whose precise attribution is not straightforward, though some advances in standardization are paving the way toward it [12]. However, in any case, the power consumption will be dependent on the number of active physical processing units (e.g., P4 switches [13] or cores); the latter can be mapped to servers in the queueing model that represents a given VNF or PNF. Therefore, putting processing units 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 affecting performance. The relation between power consumption and the number of active cores in multi-core processors has been investigated, among others, in [14], where it is shown to be roughly proportional (or piecewise linear) with respect to the number of active cores at a given operating frequency. This correspondence between active cores and power consumption is the motivation behind our work in the present paper, where we want to evaluate the effectiveness of an analytical queueing model to implement a simple approach for the reduction of energy consumption, based on the determination of the minimum number of active processing units capable of maintaining a given upper bound on the average packet latency.

The paper is organized as follows. We introduce the traffic models at all levels (packet, burst and flow) in the next section, along with a simplifying hypothesis on their mutual interaction. A simple control strategy to trade off delay performance and energy saving, by switching on and off processing units is outlined in Section III. Section IV presents and discusses numerical results on the model behavior and on its performance when used to implement a control strategy in the presence of real traffic traces. Section V contains the conclusions and directions for future work.

## II. Traffic Modeling at Multiple Time Scales

We consider a node in the edge, either performing PNF functionalities (e.g., as a set of P4 switches) or supporting containerized VNFs, operating on up to a maximum number $N_{\text{max}}$ of processing units. Our control goal here is to determine the minimum number of active units ($N \leq N_{\text{max}}$), represented by servers in the queueing system, required to achieve a desired upper bound on average latency, represented by the waiting time in the queue. Conversely, the goal could be set to the minimization of waiting time with respect to the number of active processing units under a given power constraint, corresponding to an upper bound on the number of active units.

Modelling the node as a multi-server queue might reasonably (roughly) approximate the situation where an incoming burst is directed to the processing unit with the lowest workload (in bits) at the moment of flow arrival (the model would actually correspond to such assignment if it were done on a packet-by-packet basis, disregarding flows and bursts - see [15], pp. 166-167).

We assume in this paper that the task execution of the functionality to be performed by the PNF or the VNF on incoming packets may be represented by a deterministic service time D. This may be the case, for instance, of a User Plane Function (UPF) [16] intercepting S1 Application Protocol (S1-AP) messages and parsing their content against the information available at the Network Service Provider (NSP) Operations Support System (OSS).

We adopt a bursty traffic model, where the incoming traffic of each flow can be modelled as Poisson burst arrivals with Zipf-distributed length (in $\text{pkt}/\text{burst}$), whenever $N$ processing units are active. Then, we are in the presence of an $\text{M}^x/\text{D}/N$ queueing system, whose stationary distribution can be determined analytically [11]. It is worth noting that, in the case of a single traffic class (i.e., traffic generated even by different flows, but with the same statistical characteristics, whose bursts are superposed) under deterministic service times, this model guarantees the conservation of packet/task ordering at the receiving end. We suppose further that the bursts being multiplexed in the queue are generated by a number $M$ of flows, whose statistical distribution is given by a birth-death model: flows are generated by a Poisson distribution with parameter $\lambda_f$ and have an exponentially distributed duration with parameter $\mu_f$. The flow in this case can represent the minimum granularity adopted by the network operator to group, according to some criterion, requests and data from UEs that are forwarded to a specific micro data center in the edge. All the notations are summarized in Table I.

Let $\beta$ be the average burst length in packets, $\lambda$ the burst arrival rate per flow, and $M$ the random variable representing the number of active flows. Then, conditioning to a realization $m$ of the number of active flows $M$, the total packet generation rate would be $m\lambda\beta$, and the queue utilization factor would be given by

$$\rho\left(m\right) = \frac{m\lambda\beta D}{N} \tag{1}$$

Note that the stability of the queue would impose an upper bound on the number of active flows at any time, which could be obtained by enforcing an admission control to the flows accessing the queue, so that

$$\frac{m\lambda\beta D}{N} < 1 \implies m < \lfloor \frac{N}{\lambda\beta D} \rfloor := m_{\text{max}} \tag{2}$$

where $\lfloor x \rfloor$ represents the highest integer less than or equal to $x$.

TABLE I
TABLE NOTATION

| Symbol | Description |
|---|---|
| $N$ | Number of active processing units |
| $N_{\text{min}}$ $N_{\text{max}}$ | Minimum and maximum number of processing units |
| $M$ | Random variable representing the number of active flows |
| $\lambda_f$ $1/\mu_f$ | Average rate and average duration of flow birth-death model |
| $\beta$ | Average burst length in packets |
| $\lambda$ | Burst arrival rate per flow |
| $X$ | Random variable representing the burst size |
| $\rho\left(m\right)$ | Utilization factor |
| $m$ | A realization of the number of active flows |
| $L_q\left(m\right)$ $W_q\left(m\right)$ | Average queue length and waiting time of the $\text{M}^x/\text{D}/N$ system with $m$ active flows |
| $p_j$ | Probability of having $j$ packets in the queueing system |
| $\pi_m$ | Probability that $m$ flows are active (producing bursts) on the queue in front of the processing units |
| $P_B$ | Blocking probability |
| $D$ $\mu$ | Packet service time and rate |
| $A_f$ | Traffic intensity of the flows |
| $\eta$ | Timescale of the simulations |
| $\lambda_p$ | Aggregate incoming rate ($\text{Mpkts/s}$) |
| $\Psi$ | Average Throughput (in $\text{Mpkts/s}$) |
| $\overline{W}^*$ | Desired upper bound on the average queueing delay |
| $\overline{W}_q$ | Average packet latency |
| $C_0$ $C_7$ | PU power states used in the model according to the ACPI specification [17] |

Following [11], the average queue length $L_q\left(m\right)$ of the $\text{M}^x/\text{D}/N$ system with $m$ active flows can be written analytically as

$$L_q\left(m\right) = \frac{1}{2N\left[1-\rho\left(m\right)\right]}\Bigg\{\left[N\rho\left(m\right)\right]^2 - N\left(N-1\right) + \sum_{j=2}^{N-2}\left(N\left(N-1\right)-j\left(j-1\right)\right]p_j + N\rho\left(m\right)\left(\frac{\mathbb{E}\left\{X^2\right\}}{\beta}-1\right)\Bigg\} \tag{3}$$

where $X$ is the random variable representing the burst size and $p_j, j = 2, \ldots, N-2$ are the probabilities of having $j$ packets in the queueing system. The latter can be calculated numerically by using the Probability Generating Function (PGF) of the queue length distribution and the inverse Discrete Fast Fourier Transform (iDFFT) – see [11], p. 396 and Appendices G and D). Then, by applying Little's Theorem, the average waiting time is given by

$$W_q\left(m\right) = \frac{L_q\left(m\right)}{m\lambda\beta} \tag{4}$$

The reason for computing the average waiting time conditional to a given number of flows stems from the fact that, as the time scales at the burst- and flow-level are widely different, it makes sense to consider that variations in the number of flows would occur on a much longer time scale with respect to that corresponding to variations in the number of packets in the queue. Based on this consideration, we can ignore non-stationary behaviours, and assume that a stationary state in the queue probabilities can be reached between birth and death events at the flow level (a precise treatment, based on Courtois' decomposition, of a somehow related problem can be found in [18]).

Then, we can further average out the average waiting time provided by the stationary distribution of our queuing model with respect to the flow generation; in doing so, we should take into account the upper limit in Eq. (2). Let $A_f = \lambda_f/\mu_f$ [Erlangs] denote the traffic intensity of the flows. Then, the probability $\pi_m$ that $m$ flows are active (producing bursts) on the queue in front of the processing units is given by the stationary distribution of a $\text{M}/\text{M}/m_{\text{max}}/m_{\text{max}}$ queueing system:

$$\pi_m = \text{Prob}\left\{M=m\right\} = \pi_0\prod_{j=0}^{m-1}\frac{A_f^j}{j+1} = \frac{A_f^m/m!}{\sum_{j=0}^{m_{\text{max}}}A_f^j/j!} \tag{5}$$

Finally (by conditioning to the presence of at least one active flow), the unconditional average packet latency can be written as

$$\overline{W}_q = \frac{1}{1-\pi_0}\sum_{m=1}^{m_{\text{max}}}W_q\left(m\right)\pi_m \tag{6}$$

## III. PERFORMANCE AND ENERGY SAVING TRADEOFF

The model just described refers to a single PNF or VNF processing cluster (characterized by a single queue) of processing units where incoming flows have been diverted. By exploiting the closed-form expressions obtained, we can formulate a simple optimization problem with respect to the tradeoff between the number of active processing units (which is related to the energy consumption, according to the evaluations in [14]) and queueing delay. Specifically, for all admissible offered load values $\lambda\beta$ we can state the following

Problem: Let $\overline{W}^*$ be a desired upper bound on the average queueing delay. We want to determine the minimum number of active processors that guarantees the upper bound, i.e.,

$$\overline{W}_q \leq \overline{W}^* \tag{7}$$

The admissible offered load values are those determined by a strict criterion such as that imposed on $\lambda\beta$ by the satisfaction of Eq. (2) for all possible values of $N/D$ with $1 \leq N \leq N_{\text{max}}$ i.e.,

$$\left\{\lambda\beta : \frac{m\lambda\beta D}{N} < 1 \qquad \forall m = 1, \ldots, m_{\text{max}}\right\} \tag{8}$$

We note, in passing, that the case of heterogeneous flows, both in terms of statistical characterization of parameters and of performance requirements, is more complex but still analytically tractable under a suitable setting. As was already

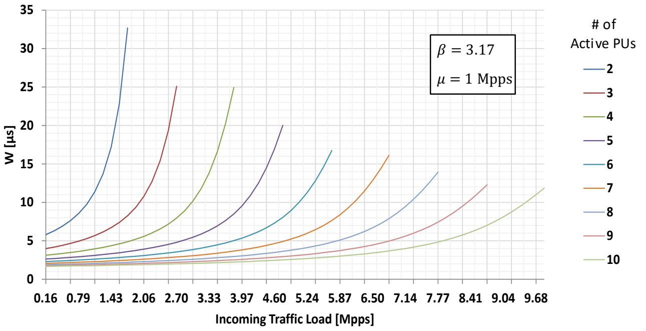

Fig. 1. Average packet latency for the Single-Flow Multi-Server Queuing Model with $\beta = 3.17\,\mathrm{pkts/burst}$ and $\mu = 1\,\mathrm{Mpkts/s}$, by varying number of active PUs.

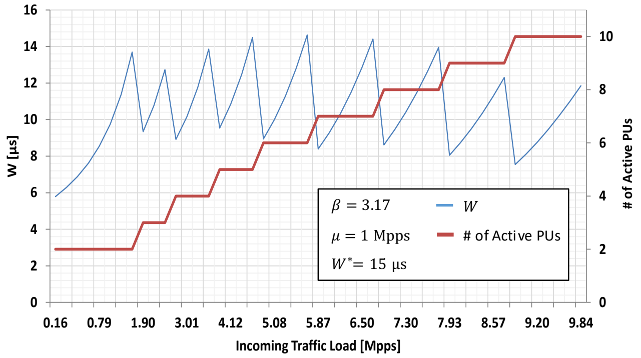

Fig. 2. Average packet latency and number of active PUs for the Single-Flow Multi-Server Queuing Model with optimization procedure, with $\overline{W}^* = 15\,\mu\mathrm{s}$, $\beta = 3.17\,\mathrm{pkts/burst}$ and $\mu = 1\,\mathrm{Mpkts/s}$.

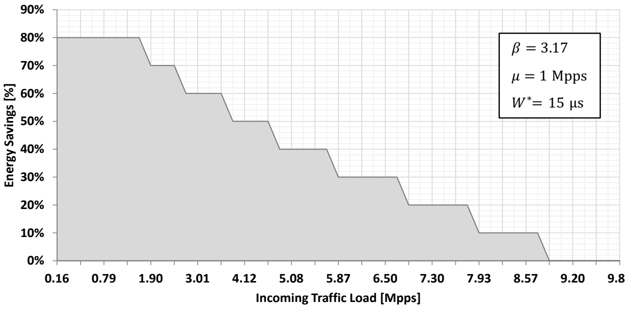

Fig. 3. Average potential in energy saving for the Single-Flow Multi-Server Queuing Model with $\overline{W}^* = 15\,\mu\mathrm{s}$, $\beta = 3.17\,\mathrm{pkts/burst}$ and $\mu = 1\,\mathrm{Mpkts/s}$.

noted in [19] in a slightly different context, in this case the most advisable and manageable model would be that of service separation [20], whereby only flows with the same statistical characteristics are multiplexed together and feed the same buffer with their bursts.

## IV. Numerical Results

We analyze here various numerical results for the determination of the minimum number of active processing units that guarantees the latency upper bound. We first compute the average packet latency time $W_q\,(m)$ at $m = 1$ (a single incoming flow) as a function of the offered load $\lambda\beta$, by varying the number of active PUs $N$ (servers in our model) from 2 to 10. For each server, we fix the packet service rate $\mu = 1\,\mathrm{Mpkts/s}$. Hence, the maximum packet service rate is $\mu N = 10\,\mathrm{Mpkts/s}$ when all servers are active, while the packet service time of our model is $D = 1/\mu = 1\,\mu\mathrm{s}$. We consider three different values for the average burst length $\beta$; namely, 2.06, 3.17 and 5.26.

In the single flow case, the packets are uniformly distributed among the $N$ servers. Figure 1 shows the average packet latency $W_q\,(1)$ (indicated as $W$ for simplicity) for the value of the parameter $\beta = 3.17\,\mathrm{pkts/burst}$. We do not report the other cases, as the results show no significant differences with respect to variations of the parameter $\beta$; with all the three values of $\beta$, $W$ is always lower than $35\,\mu\mathrm{s}$.

With a desired upper bound on the queueing delay $\overline{W}^* = 15\,\mu\mathrm{s}$, the application of the simple rule of activating the minimum number of processing units capable of maintaining it is shown in Figure 2, always with respect to $\beta = 3.17\,\mathrm{pkts/burst}$. The potential in energy saving is reported in Figure 3, by simply evaluating $(N_{\max} - N)\,/N_{\max}$ for the case $N_{\max} = 10$. The cases corresponding to the other values of $\beta$ present a similar trend.

Regarding the case of stochastic flows (with exponentially-distributed inter-arrival times and duration, following Eq. (5) for the probability distribution of the number of active ones), the average packet latency time $\overline{W}_q$ (again indicated by $W$ in the graphs) is computed by varying the value of the traffic intensity $A_f$ in Erlangs and by averaging over the distribution of active flows. Regarding the parameter $\lambda$, we fixed its value equal to $\mu/\beta$ (i.e., the upper bound to guarantee stability with a single active processor). Figure 4 shows the trend of $\overline{W}_q$, with $\beta = 3.17\,\mathrm{pkts/burst}$ and $\mu = 1\,\mathrm{Mpkts/s}$. In this case, the values are higher than those in the Single-Flow case. The delay per packet is averaged over variable active flows, which can range from 1 up to the maximum allowed for stability.

Figure 5 shows the Average Throughput $\Psi$ (in $\mathrm{Mpkts/s}$) computed by using the blocking probability $P_B = \pi_{m_{\max}}$ stemming from Eq. (5), i.e.

$$\Psi = \lambda\beta A_f\,(1 - P_B) \tag{9}$$

Finally, Figures 6 and 7 show the results of the optimized procedure with $\overline{W}^* = 75\,\mu\mathrm{s}$, $\mu = 1\,\mathrm{Mpps}$ and $\beta = 3.17\,\mathrm{pkts/burst}$. When the average incoming load is less than $1.30\,\mathrm{Mpkts/s}$, it is sufficient to have only 2 PUs active. With the increase of the incoming traffic load, the number of active processors necessary to maintain the latency below $75\,\mu\mathrm{s}$ grows (stepwise) almost linearly.

In order to validate the proposed model, we performed an extensive simulation campaign to use the results as a term of comparison. We used real-world traffic traces that are publicly available in [21], increasing the traffic volumes in the original trace by a scaling factor of 100. The incoming traffic load has an average value around $5\,\mathrm{Mpkts/s}$ with maximum values

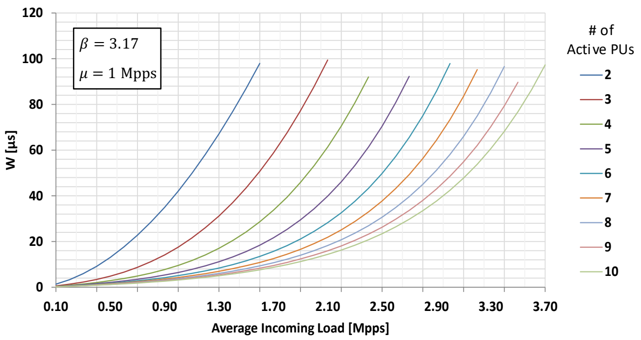

Fig. 4. Average packet latency for the Multi-Flow Multi-Server Queuing Model with $\beta = 3.17$ pkts/burst and $\mu = 1$ Mpkts/s by varying number of active PU.

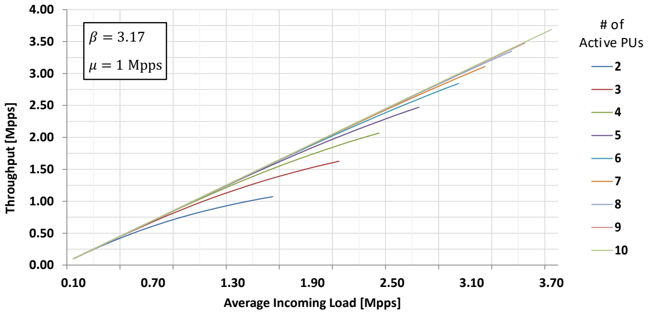

Fig. 5. Average Throughput (in Mpkts/s) for the Multi-Flow Multi-Server Queuing Model with $\beta = 3.17$ burst and $\mu = 1$ Mpkts/s, by varying number of active PU.

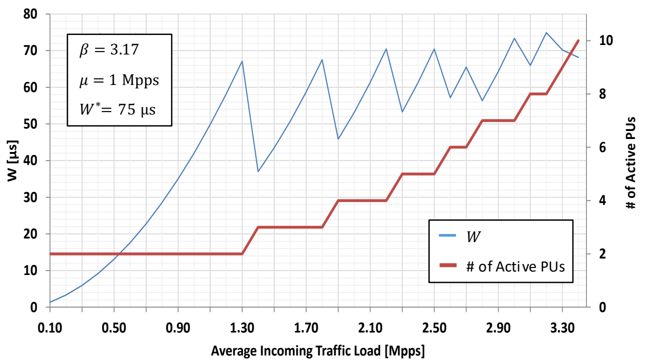

Fig. 6. Average packet latency and number of active PU for the Multi-Flow Multi-Server Queuing Model with optimization procedure, with $\overline{W}^* = 75\,\mu$s, $\beta = 3.17$ pkts/burst and $\mu = 1$ Mpkts/s.

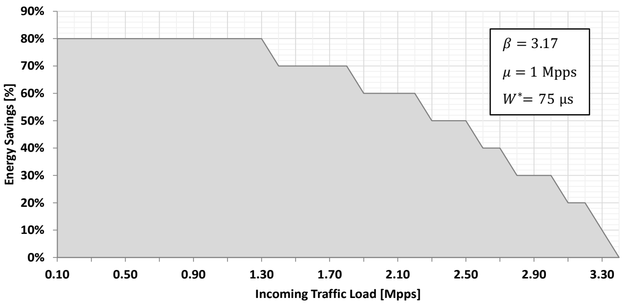

Fig. 7. Average energy saving (in %) for the Multi-Flow Multi-Server Queuing Model with $\overline{W}^* = 75\,\mu$s, $\beta = 3.17$ pkts/burst and $\mu = 1$ Mpkts/s.

of almost 10 Mpkts/s; the average value of the burst length approximately equals 3 pkt/burst. In order to save energy by turning on/off the PUs it is necessary to implement an optimization procedure that selects the minimum number of active PUs necessary to ensure a given level of performance.

We performed our simulations considering three different timescales (indicated by $\eta$): 10 s, 60 s and 300 s. The optimization procedure uses a static table, which presents different values of the pair $\lambda_p$, $N_{min}$, where $\lambda_p$ is the aggregate incoming rate (Mpkts/s) that can be supported by turning "*on*" $N_{min}$ PUs, while guaranteeing at the same time that the average packet latency is below a given performance cap ($\overline{W}^* = 90\,\mu$s). An example of static table is shown in Table II.

In this way, the optimization procedure uses the estimates of $\beta$ and $\lambda$ to select the correct value of $\lambda_p$ and the relative minimum number of active PU $N_{min}$ necessary to serve the specific arrival packet rate within the desired average delay. We fixed the total number of PUs $N_{\max} = 10$. Each PU is modelled by using only two possible power states – according to the ACPI specification [17]:

- $C_0$, where the PU performs packet processing and it is able to process 1 Mpkts/s. The power consumption in this state is 35 W.
- $C_7$, where the PU does not perform any packet processing (we considered it as in an "*off*" state) and the relative power consumption is 26.6 W.

The transition time necessary to wake-up and put to sleep a PU (from $C_0$ to $C_7$ and vice versa) is set to 140 ms.

Figure 8 shows the comparison between the average packet latency computed by the model and the one measured with the simulation. The results outline how the error (in %) between the model and the simulation is, on average, slightly higher than 5 %. In addition, there is an underestimation of the latency by the model. This underestimation could lead to a too aggressive policy (from the point of view of energy-conservation) by the optimization procedure. Indeed, it may leave a number of active PUs too low to ensure the desired level of performance. To overcome this problem, it is possible to set the latency cap $\overline{W}^*$ to a more conservative value.

Figure 9 shows the potential for energy saving, which is computed considering the relative number of active PUs with respect to the total number of PUs, whose average value is about 25 %.

The results for the other values of $\eta$ (not shown) exhibit a similar behaviour. However, already with $\eta = 60$ s, a smaller number of wake-up and sleep events of the PUs is observed, owing to the fact that the time horizon of the optimization procedure is higher than in the previous case. Hence, in order to respect the performance constraint, the

TABLE II

CORRESPONDENCE BETWEEN INPUT RATE AND NUMBER OF PUS.

| Range of $\lambda_p$ (Mpkts/s) | $N_{min}$ |
|---|---|
| $0 < \lambda_p < 4.2$ | 5 |
| $4.2 \leq \lambda_p < 5.6$ | 6 |
| $5.6 \leq \lambda_p < 6.5$ | 7 |
| $6.5 \leq \lambda_p < 7$ | 8 |
| $7 \leq \lambda_p < 7.5$ | 9 |
| $7.5 \leq \lambda_p < 10$ | 10 |

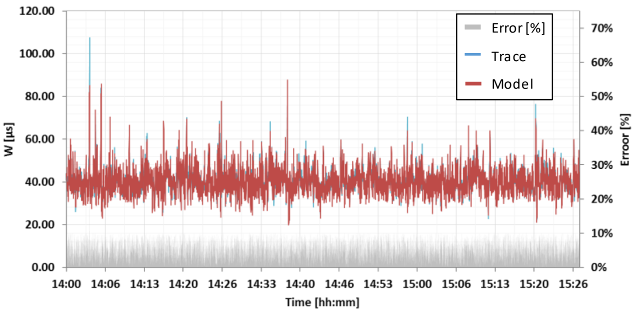

Fig. 8. Average packet latency estimated by model and simulation with $\eta = 10$ s.

optimization procedure might select each time a number of active PUs higher than the corresponding one of the previous cases. Nonetheless, the relative energy saving does not have significant differences compared to the case with $\eta = 10$ s. With $\eta = 300$ s the energy saving is slightly lower than in the other cases.

## V. CONCLUSIONS

We have considered the problem of trading off energy efficiency and performance (in terms of queueing delay), of PNFs implemented by dedicated processing units (which may be employed in lieu of VNFs in critical applications, e.g., to offload micro data centers in delay sensitive MEC contexts) or of containerized VNFs. The effect of modulating the power consumption by switching on and off computing elements on a longer time scale with respect to that of packet processing has been investigated based on the adoption of queueing models, in the presence of multiple flows multiplexed at a processing node of this kind. Simulation with real traces, from which model parameters have been estimated, has been employed to validate the results. Future work will be based on the application of a stochastic knapsack model, as briefly mentioned in Section III.

## REFERENCES

[1] J. Shuja, K. Bilal *et al.*, "Survey of Techniques and Architectures for Designing Energy-Efficient Data Centers," *IEEE Systems Journal*, vol. 10, no. 2, pp. 507–519, 2016.

[2] M. Ismail, W. Zhuang *et al.*, "A Survey on Green Mobile Networking: From The Perspectives of Network Operators and Mobile Users," *IEEE Comm. Surv. & Tuts*, vol. 17, no. 3, pp. 1535–1556, 2014.

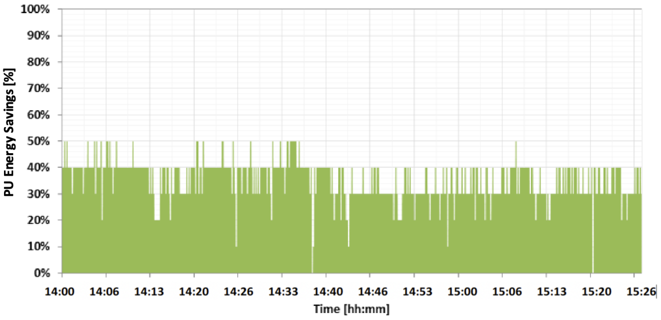

Fig. 9. Potential for energy saving based on the execution of the optimized procedure with $\eta = 10$ s on real traffic trace.

[3] R. Bolla, R. Bruschi *et al.*, "Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures," *IEEE Comm. Surv. & Tuts.*, vol. 13, no. 2, pp. 223–244, 2011.

[4] D. Kreutz, F. M. V. Ramos *et al.*, "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[5] R. Mijumbi, J. Serrat *et al.*, "Network Function Virtualization: State-of-the-Art and Research Challenges," *IEEE Comm. Surv. & Tuts.*, vol. 18, no. 1, pp. 236–262, 2015.

[6] G. Faraci and G. Schembra, "An Analytical Model to Design and Manage a Green SDN/NFV CPE Node," *IEEE Trans. on Netw. and Serv. Mngmnt.*, vol. 12, no. 3, pp. 435–450, 2015.

[7] I. F. Akyildiz, A. Kak, and S. Nie, "6G and Beyond: The Future of Wireless Communications Systems," *IEEE Access*, vol. 8, pp. 133 995–134 030, 2020.

[8] D. Qi, S. Shen, and G. Wang, "Virtualized Network Function Consolidation Based on Multiple Status Characteristics," *IEEE Access*, vol. 7, pp. 59 665–59 679, 2019.

[9] R. Bolla, R. Bruschi *et al.*, "The Dark Side of Network Functions Virtualization: A Perspective on the Technological Sustainability," in *2017 IEEE Int. Conf. Comm. (ICC)*, 2017, pp. 1–7.

[10] R. Zoppoli, M. Sanguineti *et al.*, *Neural Approximations for Optimal Control and Decision*. Cham, Switzerland: Springer Nature, 2020.

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

[12] ETSI, "EE; GAL; power management capabilities of the future energy telecommunication fixed network nodes. enhanced interface for power management in NFV environments," ETSI, Tech. Rep. ES 203 682, October 2019.

[13] P. Bosshart, D. Daly *et al.*, "P4: Programming Protocol-Independent Packet Processors," *ACM SIGCOMM Comp. Comm. Rev.*, vol. 44, no. 3, pp. 87–95, 2014.

[14] I. Takouna, W. Dawoud, and C. Meinel, "Accurate mutlicore processor power models for power-aware resource management," in *2011 IEEE Ninth Int. Conf. on Dep., Aut. and Secure Comp.*, 2011, pp. 419–426.

[15] D. Bertsekas and R. Gallager, *Data Networks (2nd Ed.)*. Englewood Cliffs, NJ, USA: Prentice-Hall, Inc., 1992.

[16] 3GPP, "System Architecture for the 5G System," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 23.501, 04 2019, version 16.0.2.

[17] ACPI, "Advanced Configuration and Power Interface Specification," 2000. [Online]. Available: http://www.acpi.info/

[18] S. Ghani and M. Schwartz, "A Decomposition Approximation for the Analysis of Voice/Data Integration," *IEEE Trans. on Comm.*, vol. 42, no. 7, pp. 2441–2452, 1994.

[19] F. Davoli, M. Marchese, and F. Patrone, "Flow Assignment in Multi-Core Network Processors," in *Advances in Optimization and Decision Science for Society, Services and Enterprises, ODS, Genoa, Italy, Sept. 2019, AIRO Springer Series*, 2020, pp. 493–503.

[20] K. W. Ross, *Multiservice Loss Models for Broadband Telecommunication Networks*. Berlin, Heidelberg: Springer-Verlag, 1995.

[21] MAWI, "(Measurement and Analysis on the WIDE Internet), Working Group Traffic Archive, Sample Point F." [Online]. Available: http://mawi.wide.ad.jp/mawi/samplepoint-F/2013