Flow Assignment in Multi-Core Network Processors

Franco Davoli[*,°], **Mario Marchese**[*,°], **Fabio Patrone**[*,°]

[*]Department of Electrical, Electronic and Telecommunications Engineering, and Naval Architecture (DITEN), University of Genoa, Italy

[°]National Laboratory of Smart and Secure Networks (S2N), National Inter-university Consortium for Telecommunications (CNIT), Genoa, Italy

Abstract In modern telecommunication networks, the trend toward "softwarization" is shifting the execution of switching and protocol functionalities from specialized devices to general purpose hardware located in datacenters or at the network edge. Incoming flows generated by User Equipment are processed by different functional modules executed in Virtual Machines (VMs) or containers. The paper considers a modeling and control architecture in this environment, for the assignment of flows to the first functional blocks in a chain of Virtual Network Functions (VNFs) and the balancing of the load among the VMs where they are executed.

Introduction

Telecommunication networks are undergoing a profound evolution, which is bringing part of their infrastructure ever closer to that of computing systems. With the advent of Software Defined Networking (SDN) [1] and Network Functions Virtualization (NFV) [2], Network Service Providers (NSPs) have started considering an increasing level of "softwarization" of the functionalities to be performed, especially as regards the access segment [3]. This trend has been further strengthened by Mobile Edge Computing (MEC) [4], [5], and by the consolidation of the fifth generation of mobile networks (5G) [6], providing a much stronger integration between the wireless mobile access and the fixed transport network and enhancing configuration flexibility through the concept of network slicing [7].

In this scenario, more and more often resource allocation and network control problems are encountered that present analogies with similar settings in computing systems and datacenters. Typically, given a set of general-purpose computing machinery, deployed by an Infrastructure Provider (InP) – or by the NSP itself over

the networking infrastructure of the InP – they will host multiple tenants that act as NSPs for their (fixed or mobile) customers; the latter run applications on their User Equipment (UE) that may need computing resources that are partly local (on the very same UE) and partly residing in a datacentre or at the mobile edge (with the latter subject to possible latency constraints that may require resource reallocations to follow users on the move).

What we address in this paper is the modelling and control architecture of a fairly general problem in this framework, where multiple incoming flows with Quality of Service (QoS) constraints (typically, on latency) share the computing resources of multi-core network processors, which perform some specific functionality in the form of Virtual Network Functions (VNFs) in the NFV environment. By modelling the incoming traffic generated by each flow in the form of bursts of packets, we adopt a simple but general model for the queueing systems that represent packet-level processing. On top of this, we construct an optimization scheme to implement the assignment and load balancing of incoming flows, characterized by statistical models with much longer time scales than the packet traffic they generate, to the processing queues, over time periods within which they are served with constant rates. Finally, in a hierarchical organization, where an SDN controller may decide upon a reallocation of processing speeds, the possible reallocation of the latter over the next time period could be considered.

The paper is organized as follows. We formalize our general problem in the next Section, along with the description of the control architecture in the case of homogeneous traffic. The third Section contains a formulation suitable for heterogeneous flows with different requirements. We report some preliminary numerical results based on the model in the fourth Section and the conclusions in the fifth one.

Problem statement and homogeneous flows case

We consider a queueing system as depicted in Fig. 1. The queues represent the operations performed by Virtual Machines (VMs)[1] hosting VNFs that implement some specific functionality on packets generated by the flows (representing audio/video/data streams stemming from applications running on UEs). We do not enter into any specific detail on the types of applications and network functions; our purpose here is to provide a fairly general model that could be applied to different situations by tuning the model's parameters, e.g., on the basis of available traffic traces. The service rates $R^{(1)}(t), \dots, R^{(M)}(t)$, satisfying $\sum_{i=1}^{M} R^{(i)}(t) = R(t)$, represent the amount of processing capacity dedicated to the specific VM by assigning one or multiple cores on a multi-core network processor provided or hosted by the InP, with total processing capacity $R(t)$. Each VM realizes a specific VNF instance

[1] We refer to VMs in the following, but the control architecture could be implemented with reference to containers, as well.

and, to fix ideas, we suppose them to be associated with a specific slice of a single tenant.

It is worth noting that the VMs may reside in the same physical processor, in different processors inside the same datacenter, or in different datacenters. For this reason, the assigned processing capacities may be different, and may correspond to different pricing schemes. The task of steering the traffic is performed by an SDN controller, to which the first packets of a flow are directed for classification when the flow is activated.

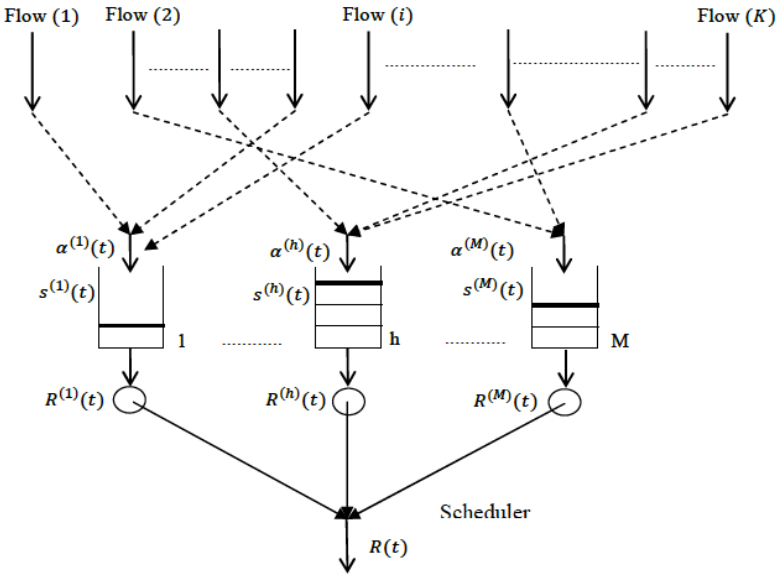

Fig. 1. Flow assignment problem.

Assuming the processing capacities $R^{(1)}(t), \dots, R^{(M)}(t)$ to have been fixed, we consider each queue with its own independent buffer in stationary conditions (and we drop the dependence on t in the following). Incoming flows are distributed among the processors on the basis of coefficients $\zeta^{(1)} > 0, \dots, \zeta^{(M)} > 0$, $\sum_{i=1}^{M} \zeta^{(i)} = 1$ (to be determined through an optimization procedure that will be described later; for the time being, they are considered fixed), in the sense that each incoming flow is assigned randomly to a processor upon its birth, according to the probability distribution determined by the coefficients.

We suppose the generation of flows to be such that each flow corresponds to a source, following a birth-death model. Packet bursts within each active flow are generated according to a Poisson model with Long-Range-Dependent (LRD) burst length. In order to take into account the traffic generation at the flow level (i.e., that the LRD traffic entering the queue is the aggregate of LRD traffic streams produced

by individual flows), for each queue i we consider the average waiting time $W^{(i)}(a^{(i)})$, $a^{(i)} = \zeta^{(i)} m\lambda\beta$ calculated by means of an $M^x/G/1$ [8] queueing model, when the aggregate burst rate is determined by the presence of m total active flows, each with burst generation rate equal to λ and average burst length β. Namely, from [8], we have

$$W^{(i)}(a^{(i)}) = \frac{{\rho^{(i)}}^2}{2\zeta^{(i)} m\lambda\beta\left(1 + \sigma^2_{S^{(i)}}/\overline{S^{(i)2}}\right)(1-\rho^{(i)})} + \frac{\rho^{(i)}\left(\overline{X^2}/\beta - 1\right)}{2\zeta^{(i)} m\lambda\beta(1-\rho^{(i)})} \quad (1)$$

where $S^{(i)}$ is the service time (depending on the distribution of the amount of operations to be performed per packet and on the processing speed $R^{(i)}$), with $E\{S^{(i)}\} = 1/\mu^{(i)}$ and mean square value and variance $\overline{S^{(i)2}}$ and $\sigma^2_{S^{(i)}}$, respectively, $\rho^{(i)} = \zeta^{(i)} m\lambda\beta/\mu^{(i)}$ is the utilization, and $\overline{X^2}$ is the mean square value of the burst length. We note, in passing, that more general models could be also considered; for instance, if energy consumption is to be included as another Key Performance Indicator (KPI) to be traded off with latency, the $M^x/G/1/SET$ could be adopted to account for set up times for processor wakeup (as done in [9], [10] in the case of deterministic service times).

Note that, for the time being, we suppose the cluster of VMs under consideration to be dedicated to serve a single class of traffic, characterized by equal generation parameters. We will extend the model to multiple classes in the next section.

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 occur on a much longer time scale with respect to that of events in the Markov chain describing the dynamics 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 is reached almost instantaneously between birth and death events at the flow level (a precise treatment of a somehow related problem can be found in [11]).

Under the above flow distribution strategy and the assumption of homogeneous flows, the same burst generation model holds for the flows being assigned to each processor. Therefore, we can examine each queue in isolation, conditioned to the presence of m total flows in the system, as an M^x/G/1 queue with input rate $\zeta^{(i)} m\lambda\beta$ [pkts/s], $i = 1, \ldots, M$. As mentioned above, the situation of flows with unequal burst generation rates (or diverse QoS requirements) will be outlined further on; however, we can already note that the more general case can be handled in a similar way if *service separation with static partitions* [12] is applied, i.e., services giving rise to flows with similar service rates and QoS requirements are grouped into classes and assigned to a subset of processors for each class.

In order to avoid instability, the following condition must be satisfied for each queue:

$$\rho^{(i)} = \zeta^{(i)} m\lambda\beta/\mu^{(i)} < 1, \text{i.e. } m^{(i)} \equiv m\zeta^{(i)} < \frac{\mu^{(i)}}{\lambda\beta} \tag{2}$$

so that the maximum number of flows $m_{\max}^{(i)}$ acceptable by queue i is equal to $\lfloor \mu^{(i)}/\lambda\beta \rfloor$.

This also imposes the presence of a Call Admission Control (CAC) on the system, such that the maximum number of flows totally acceptable be limited to

$$m_{\max} = \sum_{i=1}^{M} \left\lfloor \frac{\mu^{(i)}}{\lambda\beta} \right\rfloor \tag{3}$$

At this point, we can average out the delay over the distribution of the flows. To this aim, we suppose that both interarrival times and durations of flows can be described by independent exponential distributions, with parameters λ_f and μ_f, respectively. Let $A_f = \lambda_f/\mu_f$ [Erlangs] denote the traffic intensity of the flows. Then, the probability $p_k^{(i)}$ that k flows are active (producing bursts) on the *i*-th processor's queue is given by

$$p_k^{(i)} = \Pr\{m^{(i)} = k\} = p_0^{(i)} \prod_{j=0}^{k-1} \frac{\left(\zeta^{(i)} A_f\right)^j}{j!} = \frac{\left(\zeta^{(i)} A_f\right)^k / k!}{\sum_{j=0}^{m_{\max}^{(i)}} \frac{\left(\zeta^{(i)} A_f\right)^j}{j!}} \tag{4}$$

$k = 0,1,\ldots, m_{\max}^{(i)}$

Thus, we can write

$$\overline{W}^{(i)} = \frac{1}{\left(1 - p_0^{(i)}\right)} \sum_{k=1}^{m_{\max}^{(i)}} p_k^{(i)} W^{(i)}(\zeta^{(i)} k\lambda\beta) \tag{5}$$

for the average (with respect to the total number of flows) delay per queue (considering the presence of at least one active flow at the *i*-th VM) and

$$\overline{W} = \sum_{i=1}^{M} \overline{W}^{(i)} \zeta^{(i)} \tag{6}$$

for the total average delay over all flows. The upper limit of the sum in (5) is necessary as a consequence of condition (2).

At this point, an optimization problem can be posed for the selection of the traffic spreading coefficients as

$$\min_{\substack{\zeta^{(1)}\geq 0,\ldots,\zeta^{(M)}\geq 0 \\ \sum_{i=1}^{M}\zeta^{(i)}=1}} \overline{W} \tag{7}$$

Heterogeneous flows with different requirements

Averaging with respect to the incoming flows might be useful also in the presence of traffic with different statistical characteristics. The flow model would then correspond, in general, to a stochastic knapsack [12]. As noted, in this case the most advisable and manageable model is that of service separation, whereby only flows with the same statistical characteristics are multiplexed together and feed the same buffer with their bursts.

To fix ideas, let us suppose to have K such classes. Then, the overall processing capacity resource pool of R units can be partitioned into K groups, with R_k units assigned to the k-th group, k=1,…,K, according to some criterion. In particular, let $\theta^{(k)}(m^{(k)})$ be a function that represents the minimum processing capacity that is required to satisfy packet-level QoS requirements for $m^{(k)}$ permanent class-k flows multiplexed in a buffer. In principle, there are two possible ways to do the assignment, which we report from [12].

- *Service Separation with Static Partitions (SSSP)*: Let $R_1, \ldots, R_K$, with $R_1 + \cdots + R_K = R$, be a partition of the capacity. Under SSSP, an arriving class-k flow is admitted iff

  $$\theta^{(k)}(m^{(k)}+1) \leq R_k \tag{8}$$

  with $\theta^{(k)}(\cdot)$ corresponding, for instance, to the criteria defined by (2) or to the constraint of not exceeding a maximum average delay for the class.
- *Dynamic Partitions (DP)*. The processing capacity fractions assigned to classes are now given by $\theta^{(1)}(m^{(1)}), \ldots, \theta^{(K)}(m^{(K)})$, so that they are changing, but on a much longer time scale with respect to the packet-level dynamics. A new class-k flow would be admitted iff

  $$\theta^{(k)}(m^{(k)}+1) + \sum_{\substack{j=1 \\ j\neq k}}^{K} \theta^{(j)}(m^{(j)}) \leq R \tag{9}$$

In any case, it is interesting to note that the availability of analytical packet-level models makes relatively easy here to define a *packet level* criterion, and naturally

lends a notion of capacity of the underlying statistical multiplexer (namely, the stability preserving bound on the utilization, or the delay bound), which allows a clear definition of the flow state space.

Given the presence of a CAC, there is actually another performance index that might become of interest; namely, the blocking probability of flows (Grade of Service, GoS). The blocking probabilities at individual queues are easily calculated in the SSSP case, as done in the preceding section: the queuing model outlined above for the flow level would indeed be of type M/M/$m_{\max}^{(k)}(R_k)/m_{\max}^{(k)}(R_k), m_{\max}^{(k)}(R_k)$ being the maximum number of acceptable flows as a function of R_k, so that the blocking probabilities just correspond to the Erlang B formula, i.e.,

$$P_B^{(k)} = EB\left[\rho_f^{(k)}, m_{\max}^{(k)}(R_k)\right] = \frac{\left(\rho_f^{(k)}\right)^{m_{\max}^{(k)}(R_k)} \Big/ m_{\max}^{(k)}(R_k)!}{\sum_{j=0}^{m_{\max}^{(k)}(R_k)} \frac{\left(\rho_f^{(k)}\right)^j}{j!}} \tag{10}$$

On the other hand, in the DP case the blocking probabilities should be derived by the general stationary distribution of a stochastic knapsack.

In both situations, a general criterion could be minimizing an overall index of the type $\bar{P}_B = \sum_{k=1}^{K} P_B^{(k)}$, or $P_B^{\max} = \max_{k=1,\ldots,K} P_B^{(k)}$, with respect to the number of active processors and their allocation among classes, under given low-level constraints on delay (and, possibly, on power consumption, if we want to add this KPI to the optimization, by suitably changing the queueing models).

Numerical results

We consider an example with respect to the case of a single traffic class (homogeneous flows). To get an idea of the objective function, we plot it in the case $M = 2$, as a function of $\zeta^{(1)}, \zeta^{(2)}$, with the following numerical values of the parameters: $A_f = 10, \lambda = 20$ [burst/s], $\beta = 1.5$ [pkts/burst], $\overline{X^2} = 3$ (we have assumed a continuous approximation of the burst length, with a Pareto distribution with location parameter $\delta = 1$ and shape parameter $\alpha = 3$), $R^{(1)} = 2100000, R^{(2)} = 1600000$ [operations/s], average number of operations per packet 1000 (whence $1/\mu^{(1)} \cong 476\ \mu s, 1/\mu^{(2)} = 625\ \mu s$), $\overline{S^{(1)^2}} \cong 229408 \cdot 10^{-12}$, $\overline{S^{(2)^2}} \cong 395507 \cdot 10^{-12}$ (also here we have assumed a Pareto distribution of the service time, with shape parameter $\alpha = 10$ in both cases and location parameters $\delta^{(1)} \cong 428\ \mu s$ and $\delta^{(2)} \cong 562\ \mu s$, respectively), $\sigma_{S^{(i)}}^2 = \overline{S^{(i)^2}} - 1/\mu^{(i)^2}, i = 1,2$. The plots of the objective function are shown in Figs. 2 and 3, for the unconstrained case and over the plane $\zeta^{(1)} + \zeta^{(2)} = 1$, respectively. Fig. 4 reports the result of the optimization procedure

in this simple case, with the minimum value at $\zeta^{(1)} = 0.836$ corresponding to 271.5 μs. We have used a standard optimization tool available in the Python library (www.scipy.org), with optimization method SLSQP (Sequential Least SQuares Programming). However, it is worth noting that the form of the objective function, which is separable in the optimization variables, may suggest the use of Dynamic Programming. The possible advantages in its application will the subject of further investigation.

Considering now the case $M = 3$, we perform the optimization for a set of different values of the load generated per flow, by varying the burst arrival rate λ in the range $[10, 200]$ with discrete steps of 10 bursts/s. In this case, we have kept all the previous values, and set $R^{(3)} = 1200000$ [operations/s] $(1/\mu^{(3)} \cong 833\ \mu s,\ \delta^{(3)} \cong 750\ \mu s,\ \overline{S^{(3)^2}} \cong 703125 \cdot 10^{-12})$. The results are reported in Fig. 5, showing the tendency to a relatively stable distribution of the flows according to the processing capacities for increasing load.

Conclusions

We have considered an optimization problem in the context of multi-core network processors that provide a set of VNFs performing network operations (which may be related to network switching or MEC functionalities) on the packets generated by multiple incoming flows. The latter may be homogeneous or heterogeneous in the traffic parameters or in their requirements in terms of delay or loss. We have defined two possible optimization schemes in the two cases. Numerical results have been reported in the case of homogeneous flows. Further work will consider the numerical implementation in both cases and comparison with other assignment methods.

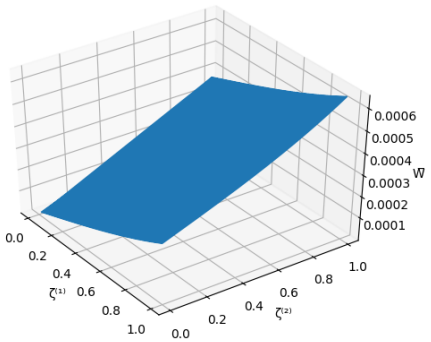

Fig. 2. Plot of the unconstrained objective function in the case $M = 2$.

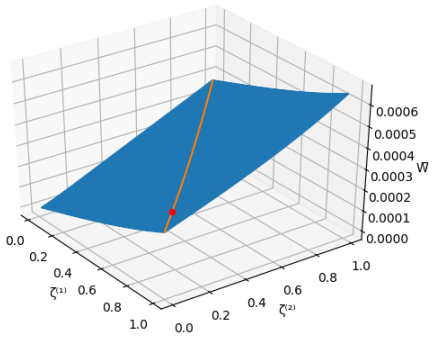

Fig. 3. Plot of the constrained objective function in the case $M = 2$.

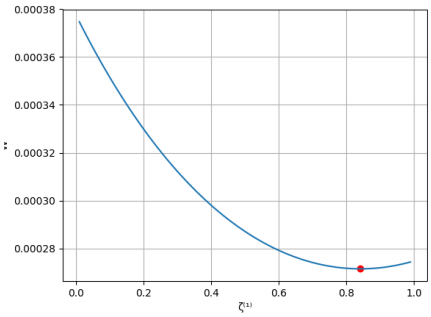

Fig. 4. Constrained cost function against $\zeta^{(1)}$ in the case $M = 2$.

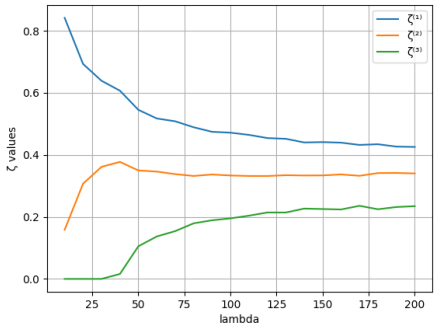

Fig. 5. Plot of the optimal allocations against the average load per flow λ [bursts/s].

Acknowledgments

This work was partially supported by the European commission, under the H2020 5G PPP project MATILDA (contract no. 761898).

References

[1] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14-76, Jan. 2015.

[2] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, R. Boutaba, "Network Function Virtualization: State-of-the-art and Research Challenges," *IEEE Commun. Surv. & Tut.*, vol. 18, no. 1, pp. 236-262, 1st Qr. 2016.

[3] A. Manzalini et al., "Towards 5G Software-Defined Ecosystems – Technical Challenges, Business Sustainability and Policy Issues," IEEE SDN Initiative Whitepaper, July 2016; http://resourcecenter.fd.ieee.org/fd/product/whitepapers/FDSDNWP0002.

[4] ETSI GS MEC 002 2016, "Mobile Edge Computing (MEC); Technical Requirements", URL: http://www.etsi.org/deliver/etsi_gs/MEC/001_099/002/01.01.01_60/gs_MEC002v010101p.pdf.

[5] "MEC Deployments in 4G and Evolution Towards 5G", ETSI White Paper, February 2018.

[6] The 3GPP Association, "System Architecture for the 5G System," 3GPP Technical Specification (TS) 23.501, Stage 2, Release 16, version 16.0.2, Apr. 2019.

[7] J. Ordóñez-Lucena et al., "Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, May 2017, pp. 80–87.

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

[9] R. Bolla, R. Bruschi, A. Carrega, F. Davoli, "Green Networking with Packet Processing Engines: Modeling and Optimization", *IEEE/ACM Trans. Netw.*, vol. 22, no. 1, pp. 110-123, Feb. 2014.

[10] R. Bolla, R. Bruschi, A. Carrega, F. Davoli, J. F. Pajo, "Corrections to: "Green Networking with Packet Processing Engines: Modeling and Optimization"", *IEEE/ACM Trans. Netw.* (to appear); published online 10 Oct. 2017, DOI: 10.1109/TNET.2017.2761892.

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

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