Towards Edge Slicing: VNF Placement Algorithms for a Dynamic & Realistic Edge Cloud Environment

Abdelquoddouss Laghrissi[1], Tarik Taleb[1], Miloud Bagaa[1] and Hannu Flinck[2]
abdelquoddouss.laghrissi@aalto.fi, tarik.taleb@aalto.fi, miloud.bagaa@aalto.fi, hannu.flinck@nokia-bell-labs.com
[1] Aalto University, Espoo, Finland
[2] Nokia Bell Labs, Espoo, Finland

***Abstract*—To support the much desired ultra-short latency of 5G mobile systems, many micro-data centers will be deployed in the vicinity of mobile users, defining a distributed edge cloud. Over this edge cloud, it is important to create optimal network slices to support different 5G verticals. Optimality is defined in terms of cost efficiency and QoS support. Therefore, it is important to understand the behavior of mobile users in terms of mobile service consumption. In this paper, we present, on one hand, a tool for developing a spatio-temporal model of mobile service usage over a particular geographical area. This tool will help to define the behavior of mobile users in terms of mobility patterns and mobile service consumption. On the other hand, based on this tool, we present a benchmark of some interesting Virtualized Network Functions (VNF) placement algorithms, among them our enhanced version of the predictive placement strategy. The comparison is based on data overload, overload of Virtual Machines (VMs) and QoS.**

I. Introduction

While many countries experience 4G with its new enhancements and advantages, and along with the wireless innovation revolution accelerating, it was easy to predict that different stakeholders would be all working to create the foundations for 5G; the 5^{th} generation mobile networks. 5G arose a big interest in the research community as it is the proposed next telecommunications standards beyond the current 4G/IMT-Advanced standards. 5G networks will support ultra-low latency, super-high throughput and massive numbers of connections. They will usher in entirely new ways to create new industries and drive unprecedented economic and societal growth. However, the most promising added value of 5G is that it will enable a fully mobile and connected society. Undoubtedly, efficient, realistic and meticulous simulation methodologies are needed.

One of the most common challenges in the simulation of such complex network scenarios is the ability to model both realistic data and behavior of each component. It is also known that the parameters, impeding the simulation of large data communication networks [1], might represent aspects of fewer significance, if not chosen wisely. In this vein, this paper introduces a tool to simulate mobile data consumption and mobility of users, based on real users' behaviors, by allowing, in the same time, the personalization of such simulation data. This personalization is enabled by choosing wisely a set of configuration inputs, related to service consumption, mobility patterns and Virtual Machine (VM) flavors. This tool is meant to become a solid ground for testing algorithms, strategies and policies for both VM and Virtual Network Function (VNF) placements. It is intended to let operators and stakeholders of communication and network sectors to have enough data that will let them understand the behavior of users in dealing with services and data consumption.

In recent years, an important development has evolved with the introduction of Network Function Virtualization (NFV) [4], [5], an architecture wherein network functions are executed over commodity servers rather than on dedicated servers. The abstraction of virtualized network elements enables the programmability of the network, increasing networking capabilities and allowing innovative service offerings, with a high cost efficiency [6], [22]. NFV and Software Defined Networking (SDN) are two of the most important enabling technologies that would be the keystone for 5G systems [23], [30]. The concept of NFV is to run network functions as software on standard VMs and through a virtualization platform. Besides NFV, SDN enables the interworking of different VNFs running on different VMs instantiated within the same data center (DC) or across multiple DCs to obtain a flexible, mobile, and dynamic network, rapidly deployable in the cloud [7], [8], [28].

Inextricably linked to NFV, the VNF placement problem has gained an important interest among researchers, for the simple reason that its outcome has a drastic impact on the NFV platform [20], [21], [27]. In this context, our paper makes two main contributions. First, we present a novel and efficient tool for spatio-temporal simulation of mobile service usage. The tool is dubbed "Network Slice Planner" (NSP). Second, we present a set of VNF placement algorithms, adding our own enhanced version of the classic predictive algorithm. The present paper is intended as an opener for future research work.

The rest of this paper is organized as follows. Section II presents the related research work existing in the literature. The NSP framework is portrayed in Section III. Section IV introduces the Advanced Predictive Placement Algorithm (APPA). The performance of the APPA scheme is compared against that of other placement strategies in Section V. Finally, the paper concludes in Section VI.

II. Related Work

To help obtaining enough data about the performance of VNF placement algorithms, user-friendly research tools are needed. These tools should support edge cloud simulations and serve as a common ground to experiment different solvers for VNF placement and VM allocation strategies, with the

978-1-5090-5019-2/17/$31.00 ©2017 IEEE

objectives to achieve cost-efficiency, QoS, etc. In the recent literature, many simulation tools have been proposed.

CloudsimNFV [24], based on CloudSim which is known for having enough extendibility to simulate NFV environment, is a NFV cloud framework intending to simulate NFV scenarios, proposing several scheduling algorithms for NFV applications. The toolkit validation and algorithm performance comparison is interesting. Unfortunately, it does not consider the most vital factor, namely the users' behavior in terms of service consumption and mobility. Many network simulators, such as NS3, NETSIM, and JSIM, are used to cope with this issue, e.g., using discrete event models which do not efficiently reflect users′ behavior.

To the best knowledge of the authors, there is no tool that offers a reliable spatio-temporal modeling of mobile service usage, mimicking at the same time realistic mobility patterns of users. Although, it is important to mention that many attempts have been carried out to understand and model the dynamics of human mobility and transportation evolution based on different mobility patterns [18], [19], in order to be able to offer better QoS to end users and make a better usage of the underlying network [25], [26]. Most of these studies mainly focus on small-scale transportation networks or rely only on either mathematical equations or visualization techniques.

VM placement is a process of mapping VMs to physical machines. As virtualization is a core technology of cloud computing, the problem of VM placement has gained lots of interest, and that is for the simple reason that the choice of VMs has a direct impact on improving power efficiency and getting the best of resource utilization. Many interesting VM placement strategies have been proposed in the recent literature. The most recent related works consider DCs' geographical location and try to find the optimal placement strategy that ensures the minimal data overload and best QoS [13], [14].

Three interesting strategies for VM placement were proposed in [13]. First, the Least Used Host (LUH) aims at ensuring load balancing: it collects the number of VMs allocated to each physical host and chooses the host that has the least number of VMs. If more than one host is at a minimum level, a random host is chosen. Second, with the same objective, N at a time (NAT) allocates N VMs at a time in a single host. If the number of VMs is a factor of N then a new host is chosen. When all hosts are packed with a factor of N, a random host is then chosen. Finally, the Least Busy Host (LBH) tries to determine the host that is least busy in terms of data traffic.

In [14], two algorithms, which are based on prediction and past VM placement decisions, have been proposed. The reactive placement (RP) algorithm considers the optimal location of application VMs to be the closest site to the majority of clients issuing requests within a given time window. It iterates through the log that contains entities for client requests, recording the origin location and data size. As new requests reach the server, they are added to the log, a list of the locations of available data centers is maintained at regular intervals. RP iterates through the window counting requests per DC location. A request is counted for a site if that site is closest to the requesting client in terms of geographical distance. The Predictive Placement Algorithm (PPA) chooses the best location of a storage VM for a given hour as the location that was closest to the majority of client accesses for that hour over the past several hours. Intuitively, this algorithm splits up a day into contiguous periods of a given number of hours, moving storage VMs at the boundary between these periods.

In this paper, besides simulating the algorithms [13], [14] using our developed NSP framework, we propose an enhanced version of the predictive placement, "Advanced Predictive Placement Algorithm" (APPA), that will be introduced in Section IV.

III. Network Slice Planning framework

As depicted in Fig. 1, the NSP framework consists of three modules, the user mobility module (UMM), the mobile edge cloud module (MECM) and the service usage module (SUM). These three modules are the pillars for VNF placement strategies, helping to determine optimal network slices. In this paper, we focus particularly on load-efficient and QoS-aware VNF placement. Each module will be detailed in the following subsections. Some interesting outputs and VM flavor choices will be also discussed.

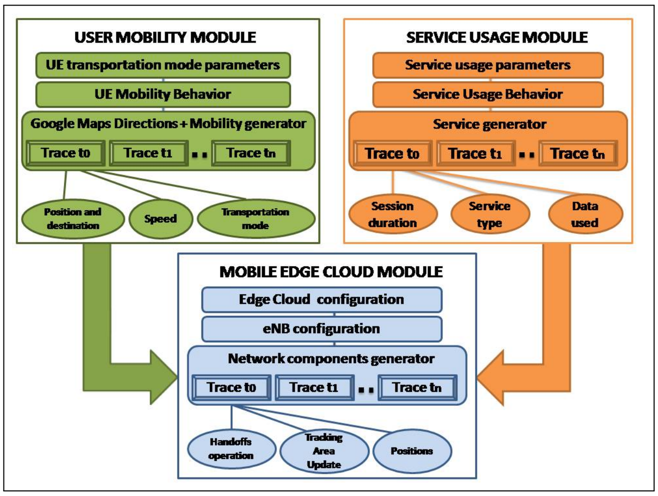

Fig. 1: The proposed NSP framework.

A. User mobility module

To define the mobility of a user, interchangeably a User Equipment (UE), the UMM is based on the logic of Google maps. It defines the distribution and mobility of UEs based on a set of itineraries. An itinerary is composed of a start and destination positions. The destination depends on the popularity of places. Popular places are known as hotspots (e.g., a university, a Mall, or a movie theater). The more popular a hotspot is, the more it attracts UEs. The starting positions and hotspots are randomly generated and can be edited and personalized manually in a dedicated page. The transportation modes of UEs are based on input parameters. In case of a moving user/UE, the itinerary can be taken in different transportation modes; walking, by bike, or by car. The mode is generated based on given mobility parameters. Then the mobility (i.e., direction, mobility path, and speed) between those two positions is obtained using Google Maps. For instance, Google Maps assumes that it takes about $15-30$ minutes to walk a mile for a person. Different mobility speeds can be found in Google Maps.

B. Service usage module

Several studies based on real datasets have been able to model the behavior of users in dealing with several services (e.g., social networks and video streaming) based on real activity log files. SUM does things backwards, by generating service sessions and requests based on those studies. In this subsection, we present the service usage model of SUM.

Initially, each UE could access a service S_k with a probability $P_{S,k}$. The set of services is composed of the following classes:

- Video streaming
- Social network
- Mobile Instant Messaging

1) Social Network Services: Based on [2], SUM models how users behave when they connect to a social network as follows. A session starts with one of the following activities:

- Browsing scrapbook
- Browsing profile of friends
- Browsing photos
- Browsing messages

When a user engages in one of these activities, he is likely to repeat the same activity but can, with a given probability, switch to a new activity within the same session, if not to a different service. During each session, several requests can be triggered. We hereafter present one of these scenarios:

- A user can access his friends$'$ profiles with probability 0.64.
- He is more likely to browse other friends$'$ profiles (0.69).
- He can access messages (0.14).
- He can logout (0.10).

The session durations and request inter-arrival times are given in [2] as follows. The inter-arrival time of the i^{th} and $(i+1)^{th}$ sessions is given as time series $a(i) = t(i+1) - t(i)$. $a(i)$ is fitted to a log-normal distribution. The probability distribution function for the log-normal distribution is given by:

$$f_1(x) = \frac{1}{\sigma x\sqrt{2\pi}} e^{\frac{-(log(x)-\mu)^2}{2\sigma^2}} \tag{1}$$

with $\mu = 2.035$ and $\sigma = 1.333$

Session lengths are highly variable when users connect to social networks. The distribution is fitted to a Zipf distribution of the form:

$$f_2(x) = \beta e^{-\alpha} \tag{2}$$

with $\beta = 3.758$ and $\alpha = 1.765$

TABLE I: Video data consumption.

Resolution	Duration (min)	Data consumption (Mb)
360p	1	3
	120	360
480p	1	5
	120	600
720p	1	10
	120	1200
1080p	1	15
	120	2000

The inter-arrival time between requests within a single session is fitted to log-normal distribution with parameters μ=1.789 and $\sigma = 2.366$. For the data usage, based on [10], SUM assumes the following:

- A single friend page request consumes $1300kB$.
- A messages page request consumes $1MB$.
- A scrapbook page request consumes $2MB$.
- A photo page request consumes $750kB$.

2) Video Streaming Services: In the following fashion [12], SUM models the video streaming service. The inter-arrival times of video streaming sessions follow a log normal distribution (1) with parameters $\mu = 2.1$ and $\sigma = 1.3$. Each video has a given line of vertical resolution and a duration. The line of vertical resolution is randomly generated. The available line of vertical resolutions for SUM are: $1080p$, $720p$, $480p$ and $360p$, where p stands for progressive scan, which is the type of video a device display uses. The resolutions $720p$ and $1080p$ generally refer to standard HD resolutions with a 1:1 pixel aspect ratio and a 16:9 display aspect ratio, respectively. The data length of a video is obtained based on the generated line of vertical resolution and the video length, which follows a power-law distribution of the form (2), with parameters β = 4.6 and α = 1.53. As in [10], the video data consumption is given as shown in TABLE I.

3) Instant Messaging Services: Based on real traffic measurements on a large scale cellular network [11], SUM models the mobile instant messaging service as follows:

- The inter-arrival time of messages can be characterized by a log normal distribution (1) with parameters $\mu = 2.245$ and $\sigma = 1.133$.
- The message length can be characterized by a power-law distribution (2) with parameters β = 4.888 and α = 1.765.

During the mobility of a UE, based on the service models presented above, SUM records the service usage activity, with a set of traces that contain the service type name, how much data consumed, request duration and the position given by UMM.

C. Mobile Edge Cloud module

MECM consists of two stages: $i)$ the configuration stage and $ii)$ the simulation stage. The configuration stage is when positions of edge clouds (ECs) and evolved Node Bs (eNBs), their ranges, bandwidths and other parameters are defined. In the simulation step, the events, such as hand-off operations and Tracking Area Updates (TAU), are recorded in log files during the mobility of UEs. The main configuration stage components are defined in follows.

ECs are the key components of MECM. Their initial locations are defined by latitude and longitude coordinates (lat, lon). The locations can be modified manually in a dedicated page. Each edge cloud has a set of eNBs deployed in its vicinity. Each eNB is characterized by a transmission range and a bandwidth, and belongs to a unique Tracking Area (TA). Each eNB is identified by a unique ID and unique coordinates. The Mobility Management Entity (MME) keeps records of the mobility of UEs in idle mode at the granularity of TA level. A TAU would be generated and transmitted when a UE moves from a TA to another [3].

Initially, NSP defines TAs, whereby each TA consists of eNBs. Formally, each node would be assigned initially one TA. Thus, we can assign, initially, each group of eNBs to a unique TA. A Tracking Area Identifier (TAI) and a Tracking Area Code (TAC) form the ID of a TA. A TAC is the unique code that each operator assigns to each of its TAs. A TAI consists of a Public Land Mobile Network (PLMN) ID and a TAC. A PLMN ID is a combination of a Mobile Country Code (MCC) and a Mobile Network Code (MNC). This format makes a TAI uniquely identified globally.

The network needs to have updated location information about UEs in idle state to find out in which TA a particular UE is located. Periodically, the UE in idle state sends a TAU request message to a MME even when the UE stays within a TA of the TAI list. A location is believed to be new if the UE is outside of the TAs of the list. The TA configuration can be personalized in the settings page of NSP.

D. VM flavors

Depending on the instances that UE tasks require, different VM flavors can be selected; each with different options. VM flavors depend on the type of services and applications launched. In [9], the flavors are divided into three main families: the standard flavors used typically for web services and software development, High Performance Computing (HPC) flavors for scientific applications and I/O flavors for Hadoop/Spark, non-critical databases and clustered databases. VMs in NSP can be classified based on the number of cores activated when a flavor is used (vCPUs), the VM Disk capacity and Random-access memory (RAM) capacity. The VNF placement strategies that will be embedded in NSP will choose the flavors which fit the functional requirements of each respective VNF [29].

For instance, the work presented in [2] offers a list of configurations varying from “tiny VM” (i.e., 1 vCPU, 10 GB of VM Disk and 512 MB of RAM) to “xxlarge VM” (i.e., 8 vCPUs, 160 GB of VM Disk storage and 16 GB of RAM). NSP offers the possibility to define personalized VM configurations.

E. Outputs

The outputs of NSP can be defined as follows:

- the total number of service requests,
- the total number of hand-off operations,
- the total number of TAU, and
- a log containing all the details of each service request, each hand-off operation and TAU.

VNF placement algorithms have as inputs the generated logs. The outputs of these algorithms, on which the comparison results are made, are the data overload, number of VMs overload and QoS costs induced by the placement decisions. Depending on the nature of events that occurred, the log could contain the positions of the UEs, the amount of data generated per service, the duration of each service usage, the EC in which a hand-off operation happened, the TA concerned by the updates, etc. For more information on NSP framework, the reader may refer to [17].

IV. Advanced Predictive Algorithm

Similar in spirit to [14]–[16], we define our algorithm as follows (see Algorithm 1):

For each service request, a choice of VM placement is made based on the best location of EC observed for a given period time. The best location is the location that was less used and closest to the majority of UEs. A VM is migrated if the predicted location is different from the last one observed. Otherwise, it remains the same over the next hours, the decision is based on the maximum value of the score $A_{v_k}(t)$ that is calculated as follows:

$$A_{v_k}(t) = \frac{|VM_{t_0,t}|}{\sum_{x=t_0}^{x=t} data(x)_k} \times \alpha' + \frac{\sum_{x=t_0}^{x=t} dist(x)_{connectedue,k}}{|connectedue_{t_0,t}|} \times \beta' \quad (3)$$

where:

- $|VM_{t_0,t}|$ denotes the number of created VMs in the given timespan (t_0, t).
- $\sum_{x=t_0}^{x=t} data(x)_k$ is the total amount of data consumed during the given timespan (t_0, t).
- $|connectedue_{t_0,t}|$ denotes the number of connected UEs, in the given timespan (t_0, t), to the given Edge Cloud EC_k.
- $\sum_{x=t_0}^{x=t} dist(x)_{connectedue,k}$ indicates the distance, in the given timespan (t_0, t), between the UEs and EC_k.

The conception of this formula is motivated by the fact that the APPA scheme bases the decisions on past logs, namely, the amount of data used, the connected UEs and load of particular regions; regions wherein a placement decision has to be made for a given EC_k. Hence, APPA calculates the number of VMs created for EC_k, during the time window and over the region where the VM placement decision has to be made. Then, this value is divided by the sum of the last data usage activity observed. Also, APPA calculates the sum of distances between the given EC_k and connected UE. The sum is then divided by the number of connected UEs observed. We assume that there are redundant patterns in terms of data usage and UEs mobility. α' and β' are the weights given to each objective, namely, data load and QoS. Algorithm 1 illustrates the pseudo-code for APPA, as explained above.

V. Results

In this section, we present the results of LBH [13], LUH [13], NAT [13], RP [14], PPA [14] and APPA schemes.

Algorithm 1 Advanced Predictive Placement Algorithm

```
Require:
    $\Gamma$: A set of past log traces.
    $\Omega$: A set of tasks.
    $E$: A set of available hosts.
    $\Gamma$: The input trace for prediction and it contains
    service logs and hand off operations that occurred
    in the last 24 hours.
Ensure:
    $\mathcal{H}$: Set that will handle input traces from a given
    $t_0$ to $t$.
    $e(t, \omega)$: Host chosen to handle task $\omega$ at t
    $A_{v_E}(t)$: Set of average score observed in each host
    of E
    at t in $\Gamma$
    AL: Set of VM allocations
for all $\omega_i \in \Omega$ do
    $\mathcal{H} = \emptyset$;
    // Each time a new request arrives
    $H_{\omega_i} = \Gamma(t\omega_i - w, \omega_i)$ ;
    if $e(t_{\omega_i}, \omega_i)! = maxA_{v_E}(t_{\omega_i})$ then
        $e(t_{\omega_i}, \omega_i) = maxA_{v_E}(t_{\omega_i})$ ;
        $KeepChoice(e, t_{\omega_i}, t_{\omega_i} + w)$ ;
        $\mathcal{AL} = createAllocation(t_{\omega_i}, e)$;
        $\mathcal{AL}s = \mathcal{AL}s \cup \{\mathcal{AL}\}$;
    end if
end for
return  $\mathcal{AL}s$;
```

The inputs for the benchmarking of the VNF/VMs placement strategies are the logs of service requests recorded by NSP for 800 mobile UEs, using video streaming, instant messaging and social network services, in the region of Helsinki, for a duration of 24 hours. Here, we present results recorded during 5 hours varying from low to very high load of service requests. Comparisons and discussions will be presented on the basis of QoS, data overload and overload of VMs number.

A. QoS

QoS is based on the cost, in terms of distance and delay, between the host EC and the target client machine. The objective is to achieve the lowest cost values as shown in Fig. 2. When service requests are relatively low, the gap in terms of QoS performances is not very important. Yet, the PPA and APPA schemes show slightly better results. During peak hours when the number of service requests is high, the APPA scheme considerably outperforms the other strategies, with the PPA scheme getting in the second position of the lowest cost values.

B. Data overload

Data overload is defined as the overload of virtual disk storage used in VMs. The objective is to achieve the lowest data overload values as shown in Fig. 3. The LUH and APPA schemes outperform the other placement strategies in terms of data overload, and the gap increases as the load of services become more important. From one side, LUH gives more importance to the host that has the least number of VMs, consequently, the least used in terms of virtual disk storage. On the other side, APPA favors the least observed and closest host to most UEs.

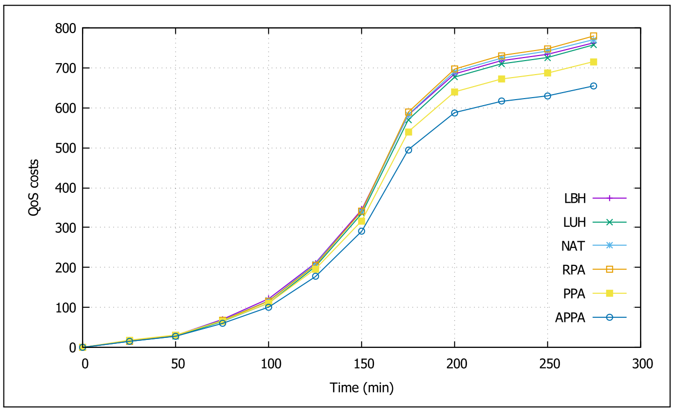

Fig. 2: QoS costs of different VNF placement strategies.

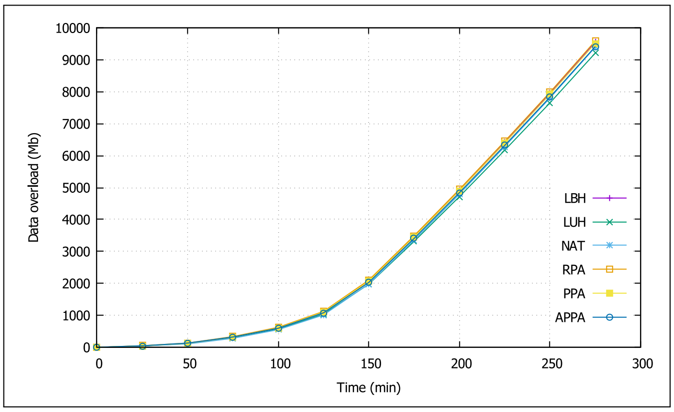

Fig. 3: Data overload of different VNF placement strategies.

C. Number of VMs overload

As the LUH strategy picks the host with the lowest number of VMs, despite the cost, it achieves the best results in balancing load among VMs compared to other strategies (see Fig. 4). NAT comes as the second best strategy to achieve load balancing among VMs, as it allocates 5 tasks to each host at a time. NAT becomes more efficient when the number of service requests is high; its variations become similar to those of LUH.

VI. CONCLUSION

In this paper, we introduced a new tool for modeling the spatio-temporal usage of mobile video streaming, mobile instant messaging and social network services, taking into account real mobility patterns of users. We also compared

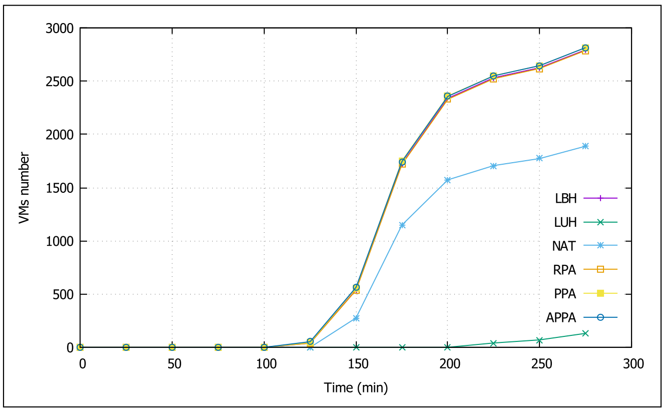

Fig. 4: VMs overload in case of different VNF placement strategies

the performance of several VNF placement strategies based on QoS in terms of delays and distance costs, data overload and on frequency of VM overload.

The paper also introduced an advanced predictive VNF placement strategy, which is an enhancement of the classic predictive VNF placement algorithm. We intend to propose a more sophisticated VNF placement strategy and improve NSP by introducing other factors such as incidents, signal disturbance and dysfunction of equipments.

Acknowledgement

This work was partially supported by the European Unions Horizon 2020 research and innovation programme under the 5G!Pagoda project with grant agreement No. 723172.

References

[1] V. Paxson and S. Floyd, "Why we dont know how to simulate the Internet," in Proc. Winter Simulation Conference, Atlanta, GA, USA, Dec 1997.

[2] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida, "Characterizing user behavior in online social networks," in Proc. 9th ACM SIGCOMM conference on Internet measurement, USA, 2009. pp. 49-62.

[3] A. Kunz, T. Taleb, and S. Schmid, "On Minimizing SGW/MME Relocations in LTE," in Proc. ACM IWCMC'10, Caen, France, Jun. 2010.

[4] N. Paper, "Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges & Call for Action. Issue 1," in Technical report, ETSI, Oct 2012.

[5] "Network Functions Virtualization (NFV): Network Operator Perspectives on Industry Progress," in ETSI, Oct 2014.

[6] A. Galis, S. Clayman, L. Mamatas, J. Rubio-Loyola, A. Manzalini, S. Kuklinski, J. Serrat, and T. Zahariadis, "Softwarization of Future Networks and Services - Programmable Enabled Networks as Next Generation Software Defined Networks," in Proc. Software Defined Networks for Future Networks and Services, Trento, Italy, Nov 2013.

[7] K. Kirkpatrick, "Software-defined networking," in Communications of the ACM, vol. 56, no. 9, Sep 2013. pp. 16-19.

[8] H. Kim and N. Feamster, "Improving network management with software defined networking," in IEEE Communications Magazine, vol. 51, no. 2, Feb 2013. pp. 114-119.

[9] *www.opensciencedatacloud.org/support/instances.html*

[10] Data usage calculator, Office of the communication authority, *http : //app1.ofca.gov.hk/apps/dataCal/data.asp*

[11] X. Zhou, Z. Zhao, R. Li, Y. Zhou, J. Palicot, and H. Zhang, "Understanding the Nature of Social Mobile Instant Messaging in Cellular Networks," in IEEE Communications Letters, vol. 18, no. 3, 2014. pp. 389-392.

[12] S. Rao, A. Legout, Y. Lim, D. Towsley, C. Barakat, and W. Dabbous, "Network characteristics of video streaming traffic," in Proc. Seventh COnference on emerging Networking EXperiments and Technologies, USA, Article 25, 2011.

[13] S. Clayman, E. Maini, A. Galis, A. Manzalini and N. Mazzocca, "The dynamic placement of virtual network functions," in IEEE Network Operations and Management Symposium, Krakow, 2014. pp. 1-9.

[14] B. Malet and P. Pietzuch, "Resource allocation across multiple cloud data centres," in Proc. the 8th International Workshop on Middleware for Grids, Clouds and e-Science, ACM, New York, NY, USA, Article 5, 2010.

[15] J. Rolia, Xiaoyun Zhu, M. Arlitt and A. Andrzejak, "Statistical service assurances for applications in utility grid environments," in Proc. 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, 2002. pp. 247-256.

[16] J. L. L. Simarro, R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, "Dynamic placement of virtual machines for cost optimization in multi-cloud environments," in Proc. International Conference on High Performance Computing & Simulation, Istanbul, 2011. pp. 1-7.

[17] http://mosaic-lab.org/implementations.aspx, NETWORK SLICE PLANNING APPLICATION section

[18] P. Wang, T. Hunter, A. M. Bayen, K. Schechtner, and M. C. Gonzalez, "Understanding road usage patterns in urban areas," in Scientific reports, 2, 2012.

[19] S. Yang, "On feature selection for traffic congestion prediction," in Transportation Research Part C: Emerging Technologies, vol 26, 2013. pp 160169.

[20] T. Taleb and A. Ksentini, "Gateway Relocation Avoidance-Aware Network Function Placement in Carrier Cloud," in Proc. ACM MSWIM 2013, Barcelona, Spain, Nov. 2013.

[21] M. Bagaa, T. Taleb, and A. Ksentini, "Service-Aware Network Function Placement for Efficient Traffic Handling in Carrier Cloud," in Proc. IEEE WCNC14, Istanbul, Turkey, Apr. 2014.

[22] T. Taleb, "Toward Carrier Cloud: Potential, Challenges, & Solutions," in IEEE Wireless Communications Magazine, Vol. 21, No. 3, Jun. 2014. pp. 80-91.

[23] T. Taleb, M. Corici, C. Parada, A. Jamakovic, S. Ruffino, G. Karagiannis, and T. Magedanz, "EASE: EPC as a Service to Ease Mobile Core Network," in IEEE Network Magazine, Vol. 29, No. 2, Mar. 2015. pp.78 88.

[24] W. Yang, M. Xu, G. Li, and W. Tian, "CloudSimNFV: modeling and simulation of energy-efficient NFV in cloud data centers," in CoRR, vol. abs/1509.05875, 2015.

[25] A. Nadembega, A. Hafid, and T. Taleb, "Mobility Prediction-aware Bandwidth Reservation Scheme for Mobile Networks," in IEEE Trans. on Vehicular Technology, Vol. 64, No. 6, Jun 2015. pp. 2561 2576.

[26] A. Nadembega, A. Hafid, and T. Taleb, "A Destination & Mobility Path Prediction Scheme for Mobile Networks," in IEEE Trans. on Vehicular Technology, Vol. 64, No. 6, Jun 2015. pp. 2577 2590.

[27] T. Taleb, M. Bagaa, and A. Ksentini, "User Mobility-Aware Virtual Network Function Placement for Virtual 5G Network Infrastructure," in Proc. IEEE ICC 2015, London, UK, Jun 2015.

[28] A. Laghrissi, S. Retal, A. Idrissi, "Modeling and optimization of the network functions placement using constraint programming," in ACM International Conference Proceeding Series, Part F126324, art. no. a52, 2016.

[29] F.Z. Yousaf and T. Taleb, "Fine Granular Resource-Aware Virtual Network Function Management for 5G Carrier Cloud," in IEEE Network Magazine, Vol. 30, No. 2, Mar 2016. pp. 110 115.

[30] T. Taleb, B. Mada, M. Corici, A. Nakao, and H. Flinck, "PERMIT: Network Slicing for Personalized 5G Mobile Telecommunications," in IEEE Communications Magazine, Vol. 55, No. 5, May 2017. pp. 88 93