Automatic Binpacking of VNF Chains for Proactive Caching

Gao Zheng, Anthony Tsiopoulos, Vasilis Friderikos
Centre for Telecommunications Research, King's College London, London WC2R 2LS, England
E-mail: {gao.zheng, anthony.tsiopoulos, vasilis.friderikos}@kcl.ac.uk

***Abstract*—Notwithstanding the significant research effort Network Function Virtualization (NVF) architectures received over the last few years little attention has been placed on optimizing proactive caching when considering it as a service chain. Since caching of popular content is envisioned to be one of the key technologies in emerging 5G networks to increase network efficiency and overall end user perceived quality of service we explicitly consider in this paper the optimization of caching based VNF service chains. To this end, we detail a mathematical programming framework tailored to VNF caching chains and also a scale-free heuristic algorithm to provide competitive solutions for large network instances since it is a variant of the classical NP-hard multi-dimensional binpacking problem. A wide set of numerical investigations are presented for characterizing the attainable system performance of the proposed schemes.**

***Index Terms*—Network Function Virtualization, network optimization, 5G networks, proactive caching, integer linear programming, heuristic algorithms**

I. INTRODUCTION

It is well accepted that current mobile network architectures suffer from insufficient scalability and flexibility to quickly accommodate new services and ability to embrace vertical industries [**?**]. To address these challenges, applying software define networking (SDN) [11] principles in emerging architectures towards 5G networks is gaining significant momentum recently [**?**]. This goes hand-in-hand With the heavily studied now network function virtualization (NFV) [2] architectures, that together with SDN, can be considered as the two enablers towards flexible wireless networks, where full virtualization and efficient network slicing according to the needs of different tenant can be implemented. An SDN/NFV-enabled network is in essence able to decouple network functions (NFs) from the underlying physical devices, thereby, NFs can be virtualized, creating the so-called virtual network functions (VNFs). The benefit is that VNFs can be flexibly controlled/assigned/moved within the network using Virtual machines or (docker) containers. In NFV framework, an end-to-end network service (e.g., rich voice/data) is described by an VNF forwarding graph, where a number of VNFs (possibly distributed in various physical nodes in the network) need to be visited in certain predefined order [**?**]. To be more precise, the sequenced VNFs of a service request form a service chaining as the service flow passes through an ingress or egress point in a virtual network device. An illustrative example of such service chain is shown in figure 1, where caching is considered as one of the VNFs[1] that constitute the overall service chain; these VNFs might be located in different nodes in the network. Our aim is to consider caching and the other possible VNFs that might be required for the service in an integrated manner in order to increase network efficiency. Undoubtedly, among different vNFs, it is expected that caching would emerge as one of the potential key network elements to be supported in emerging and future wireless/mobile networks. Viral and popular video streams dominate aggregate mobile Internet traffic[2] and it is an application well suited to various different caching strategies. In that respect, caching of popular content deserves paying a special attention in terms of VNF hosting location and chaining. This is because in the most general case, a cached content must be visited before other VNFs can be applied and this service flow might originate from different possible network locations depending on the caching strategy. Hence, the service does need to reach a gateway node but can originate at a node that host the required cached content (which can be topologically close to the end user). Therefore, the location of caches in a VNF service chain, greatly affects the overall vNF chain orchestration as well as the aggregate traffic dynamics in the network, since links of higher aggregation (deeper in the network) can reduce their utilization levels. However, efficient caching in mobile networks can be deemed as a highly challenging task since the optimality of the cache locations are dependent on the movement/mobility patterns of the users. Notably, to significantly reduce access delays to highly popular content caching content close to the end user without considering the effect of mobility might lead

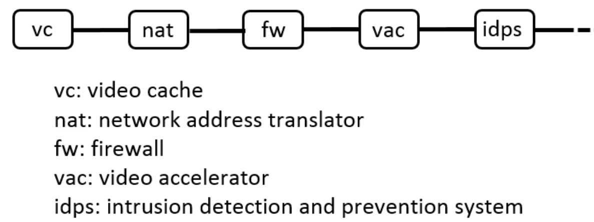

Fig. 1: An example of caching in conjunction with other virtual network functions.

[1]The terms VNF and NF are used interchangeably in the rest of the paper, except where differentiation is required.

[2]Mobile video traffic accounted for 55 percent of total mobile data traffic in 2015 according to the CISCO Global Mobile Data Traffic Forecast Update that has been released in February 2016.

to degradation of performance. In this case, caching popular content closer to the end user might inevitably require more frequently changes of the cache location to keep providing optimal performance. In this case, the caching location and the associated vNF chaining need to be jointly considered to avoid sub-optimal cases, especially under congestion episodes where performance can be significantly affected. To summarize, the focus and motivation of the paper is on enhancing proactive caching policies by taking into account the whole VNF chain.

A. Motivation and Illustrative Examples

In this paper, we propose a Proactive caching-chaining (PCC) scheme to enhance the mobility support of SDN-enabled/NFV service chaining in mobile networks. To motivate the research we discuss illustrative examples of the cross issues between caching and VNF chaining that aim to shed further light on some of the key challenges. To start with, Figure 2 shows the case of a service with two NF where the first one is caching and the other one is assumed to be a video acceleration network function. As can be seen from the figure, Case I entails a sub-optimal allocation when mobility is also taken into account. Case II shows a more suitable NF location where after the mobility event the cache and chain location is topologically closer to the end user; in Case II the NFs are located 3 hops away from the end user after the mobility event whereas in Case I, which a mobility oblivious allocation the NFs are located 4 hops away. Figure 3 shows the case where VNF chaining and pro-active caching take place independently. The figure shows potential pro-active caching locations but not in all of those pre-selected locations from the caching algorithm it is possible to host the other NFs due to numerous reasons such as for example reservation policies, placement based on affinity and/or anti-affinity rules and overall resource usage of the virtual machines [9]; for example only in one of those locations the two NF can be co-located (node b). Furthermore, as shown in figure 4 the optimal location of caching and the other NF in the service chain might be different; in the figure shown the optimal location of caching is in node (b) whereas the NF for video acceleration is located at node (d). It is therefore important firstly to consider the issue of caching and service chaining in a holistic integrated manner and secondly to optimize the location and chaining of the different NF in order to increase overall network performance.

Based on the above discussion, the proposed scheme proactively performs caching and VNF chaining so that overall network performance is optimized whilst end user receive their requests seamlessly. Notably, we take VNF chaining allocation and proactive caching as a joint problem and formulate it as a Integer Linear Programming (ILP) problem that minimize the combine cost of VNF placement, chaining and routing. We also investigate the performance obtained of a proposed scale-free heuristic algorithm since the problem resembles the NP-hard binpacking optimization problem.

In summary, we hereafter make the following key contributions, We firstly, propose a novel VNF chaining placement scheme, namely, proactive caching-chaining (PCC) that improves the mobility support for the up coming SDN-enabled/NFV network framework. Furthermore, we model and formulate the joint proactive caching and chaining problem to obtain optimal routing and placement cost and based on that we devise a scalable heuristic approach and evaluate the performance of the system.

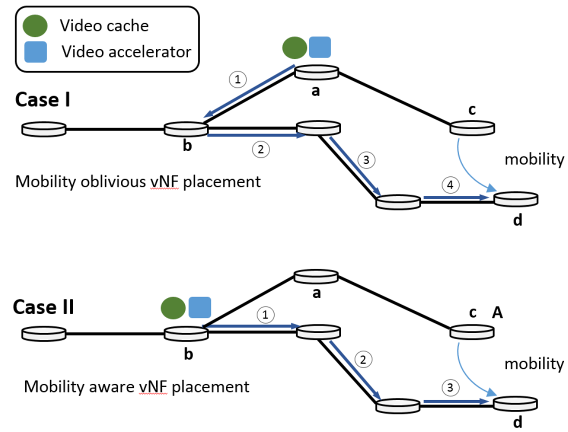

Fig. 2: Effect of mobility on the joint caching VNF chaining problem.

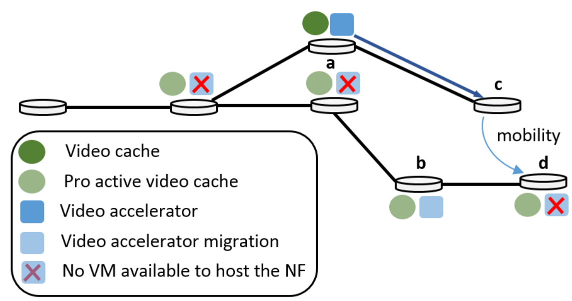

Fig. 3: Limited availability of resources (in terms of Virtual Machines for example) in the candidate pro-active caching locations to host the required VNFs for the service.

II. Previous Research Work

The overall logical architecture of the so-called VNF Management and Orchestration (MANO) architecture has been

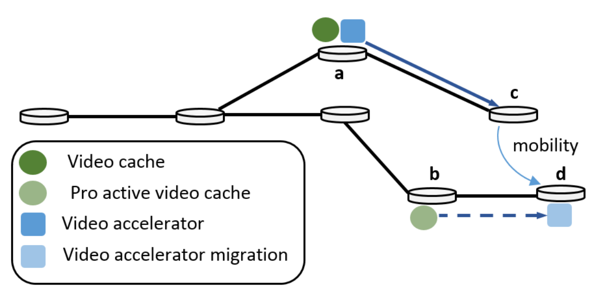

Fig. 4: An optimal VNF chain NF are located in different nodes in the network.

mainly an industry-lead initiative and has been defined within ETSI [2]. An example of a VNF orchestrator, which is called *Stratos* is presented in [3] and is built on top of a *Floodlight*[3] controller. The work in [4] can be considered as another effort to provide orchestration between virtualized NFs especially with emphasis on issues such as Virtual Machine (VM) migration and split/merging of service flows. An overview of the challenges emerging in virtual network function scheduling is presented in [5]; in this paper the authors explain the application of SDN and NFV technologies with emphasis on backbone networks. In terms of caching there has been recently a significant amount of work. A caching scheme suitable for mobile networks that takes into account user mobility has been proposed in [6] where the idea is to predict the mobility pattern of users and opportunistically cache content along the predicted path of users. A scheme that proactively cache content using transportation and focusing on video content has been presented in [7]. The idea is to utilize the almost deterministic mobility of users in transportation systems such as trains to proactively cache popular content that the users might request upon their arrival. The ideas on proactive caching in this paper resemble more closely the work in [**?**], [15] which propose a set of mobility-aware caching schemes.

However, none of previous research works make caching decisions on a view of the whole service chain. To the best of our knowledge this is the first work to consider in an explicit and integrated manner proactive caching as part of a VNF chain. In most practical cases, this simple cache moving could lead to inefficient routing of a mobile user to receive a service. Fig 1 gives an example of the inefficient routing problem where firewall as a NF must also be visited and only cache is moved [4]. It is apparent that, in order to improve the mobility support of SDN-enabled networking, other NFs on a same VNF service chain must also be moved, with the decision of caching. A close related work can be found in [**?**] which aims to assign VNFs into given SDN-enabled networks. However, it does not take routing and location of VNFs into consideration.

III. Network Modeling and Proactive chaining with caching

A mobile network is modeled as an undirected graph $G = (N, E)$, where $\mathbf{N}$ denotes the set of nodes in the network and E denotes the set of links in the network. By $\mathbf{F}$, we denote the set of NFs and f_i represents the specific NF$_i$. Each f_i, if activated, consumes/requires some physical resources (i.e., CPU cycles, DRAM memory). We uniformly describe these resource requirements as a single column matrix u_i, meanwhile, the amount of available resources of node k, which is able to host VNFs, is denoted using the single column matrix U_k.

The term "chain" in the so-called service chaining represents the different middleboxes that the service should traverse, with a specific order, across the network using software provisioning. This is the case under the proposed NFV architecture, where new services and/or network slices can be instantiated as software-only, running on commodity hardware on top of virtual machines or containers. To provide a service request $r \in \mathbf{R}$ (with $\mathbf{R}$ we denote the set of requests) for a mobile user and/or tenant, a network function forwarding graph (VNF-FG) [18] needs to access a set of corresponding NFs that are visited in a pre-defined order (which the VNF orchrstrator should preserve). In this paper, we consider the form of service request r as the set $r = \{f_1, f_2, \cdots, f_i\}$ where the sequence express the visiting order of the different network functions. For modeling simplification reasons the corresponding relationship of a NF and its order in a request, can be represented by a binary matrix V_{ril}, as follows,

$$V_{ril} = \begin{cases} 1 & \text{if the } l^{th} \text{ NF of request } r \text{ is NF}_i. \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Hereafter, we consider the scenario where a mobile user and/or tenant connected to node o and requesting $\mathbf{R}$ services. As presented in Figure 2, caching as a NF, is the head of a service request chain and it is denoted as f_0. We define a candidate node set $\mathbf{K} \subseteq \mathbf{N}$ that consist of the potential candidate nodes of hosting NFs. By $\mathbf{D}$, we define a set of potential destinations that mobile users might move due to their inherent mobility. Using historical data available to mobile network providers it is feasible to estimate such probabilities of end users moving from their current location to an a adjacent candidate destination node d. We denote this probability of changing their serving access router with ρ_d. As eluded, we assume that ρ_d is predefined by using available historical data from operators so this assumption can be deemed as realistic due to vast available data which can provide accurate characterization of user mobility patterns. With known candidate cache locations, which can be done using for example a proactive caching technique such as PCWR [15]), PCC aims to proactively place network functions $f_i \in \mathbf{F}$ into the set of nodes $\mathbf{K}$. To be more precise, we define by $\mathbf{S_r}$ to be the set of initiating nodes (i.e., proactive caching locations) of a service chain r, with $\mathbf{H}$ denoting the set of $\mathbf{S_r}$. Given $\mathbf{H}$ and $\mathbf{D}$, the proposed scheme returns the optimal proactive allocation of the NFs that minimizes the joint cost of routing, location and chaining.

A. Proactive chaining-caching problem

Based on the previously described network settings we define the following binary decision variables,

$$x_i^k = \begin{cases} 1 & \text{if NF}_i \text{ is placed at } k. \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

$$y_{ri}^{ksd} = \begin{cases} 1 & \text{if NF}_i \text{ of request } r \text{ with head } s \text{ and} \\ & \text{destination } d \text{ is visited from } k. \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

[3]www.projectfloodlight.org

[4]NF movement in this paper refers to any approach that occurs the change of the function's location. (e.g., proactive caching)

The optimal proactive chaining-caching problem is defined as the following non-linear integer optimization problem,

$$\min_{x_i^k, y_{ri}^{ksd}} \sum_{k\in\mathbf{K}}\sum_{i\in\mathbf{F}} C_i^k x_i^k + \sum_{r\in\mathbf{R}}\sum_{s\in\mathbf{S_r}}\sum_{d\in\mathbf{D}}\sum_{k\in\mathbf{K}}\sum_{i\in\mathbf{F}} \rho_d P_{sk} V_{ri1} y_{ri}^{ksd} + \sum_{r\in\mathbf{R}}\sum_{s\in\mathbf{S_r}}\sum_{d\in\mathbf{D}}\sum_{k,m\in\mathbf{K}}\sum_{i,j\in\mathbf{F}}\sum_{l=1}^{L-1} \rho_d P_{km} V_{ril} y_{ri}^{ksd} V_{rj(l+1)} y_{rj}^{msd} + \sum_{r\in\mathbf{R}}\sum_{s\in\mathbf{S_r}}\sum_{d\in\mathbf{D}}\sum_{k\in\mathbf{K}}\sum_{i\in\mathbf{F}} \rho_d P_{kd} V_{riL} y_{ri}^{ksd} \tag{4}$$

$$\text{S.t.} \sum_{i\in\mathbf{F}} u_i x_i^k \le U_k, \forall k \in \mathbf{K} \tag{4a}$$

$$\sum_{k\in\mathbf{K}}\sum_{i\in\mathbf{F}} V_{ril} y_{ri}^{ksd} \ge 1,\ \forall r \in \mathbf{R}, s \in \mathbf{S_r}, d \in \mathbf{D}, \quad l = 1, \dots L \tag{4b}$$

$$x_i^k - y_{ri}^{ksd} \le 0, \forall r \in \mathbf{R}, i \in \mathbf{F}, k \in \mathbf{K}, s \in \mathbf{S_r}, d \in \mathbf{D} \tag{4c}$$

$$x_i^k \in \{0,1\}, \quad \forall i \in \mathbf{F}, k \in \mathbf{K} \tag{4d}$$

$$y_{ri}^{ksd} \in \{0,1\}, \quad \forall r \in \mathbf{R}, i \in \mathbf{F}, k \in \mathbf{K}, s \in \mathbf{S_r}, d \in \mathbf{D} \tag{4e}$$

where C_i^k is the cost of placing NF$_i$ at k. While P_{sk}, P_{km} and P_{kd} are the shortest path routing costs between the candidate nodes. Constraint (4a) bounds the resources that can be consumed by each NF in every node. (4b) enforce that each NF in a requested chain must be visited at least once. (4C) is a binding constraint that insures the availability of a NF at a node is valid only when the NF is hosted at the node.

The first term of the objective function is the placement cost of hosting VNFs at a node. The rest of the terms in the objective function reflect the accumulative routing cost of each hop on the VNF-FG of a requested chain. To linearize the optimization problem, we replace the product of binary decision variables $y_{ri}^{ksd} y_{rj}^{msd}$ with an auxiliary variable z_{rij}^{kmsd}, which is defined as follows,

$$z_{rij}^{kmsd} = \begin{cases} 1 & \text{if request } r \text{ with head } s \text{ and destination } d \\ & \text{visits NF}_i \text{ at node } k \text{ and NF}_j \text{ at node } m. \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Hereafter, the optimization problem is converted to the integer linear programming problem shown as follows,

$$\min_{x_i^k, y_{ri}^{ksd}} \sum_{k\in\mathbf{K}}\sum_{i\in\mathbf{F}} C_i^k x_i^k + \sum_{r\in\mathbf{R}}\sum_{s\in\mathbf{S_r}}\sum_{d\in\mathbf{D}}\sum_{k\in\mathbf{K}}\sum_{i\in\mathbf{F}} \rho_d P_{sk} V_{ri1} y_{ri}^{ksd} + \sum_{r\in\mathbf{R}}\sum_{s\in\mathbf{S_r}}\sum_{d\in\mathbf{D}}\sum_{k,m\in\mathbf{K}}\sum_{i,j\in\mathbf{F}}\sum_{l=1}^{L-1} \rho_d P_{km} V_{ril} V_{rj(l+1)} z_{rij}^{kmsd} + \sum_{r\in\mathbf{R}}\sum_{s\in\mathbf{S_r}}\sum_{d\in\mathbf{D}}\sum_{k\in\mathbf{K}}\sum_{i\in\mathbf{F}} \rho_d P_{kd} V_{riL} y_{ri}^{ksd} \tag{6}$$

$$\text{S.t.} \sum_{i\in\mathbf{F}} u_i x_i^k \le U_k, \forall k \in \mathbf{K} \tag{6a}$$

$$\sum_{k\in\mathbf{K}}\sum_{i\in\mathbf{F}} V_{ril} y_{ri}^{ksd} \ge 1,\ \forall r \in \mathbf{R}, s \in \mathbf{S_r}, d \in \mathbf{D}, \quad l = 1, \dots L \tag{6b}$$

$$x_i^k - y_{ri}^{ksd} \le 0, \forall r \in \mathbf{R}, i \in \mathbf{F}, k \in \mathbf{K}, s \in \mathbf{S_r}, d \in \mathbf{D} \tag{6c}$$

$$z_{rij}^{kmsd} \le y_{ri}^{ksd}, \forall r \in \mathbf{R}, i \in \mathbf{F}, k \in \mathbf{K}, s \in \mathbf{S_r}, d \in \mathbf{D} \tag{6d}$$

$$z_{rij}^{kmsd} \le y_{rj}^{msd}, \forall r \in \mathbf{R}, i \in \mathbf{F}, k \in \mathbf{K}, s \in \mathbf{S_r}, d \in \mathbf{D} \tag{6e}$$

$$z_{rij}^{kmsd} \ge y_{ri}^{ksd} + y_{rj}^{msd} - 1, \forall r \in \mathbf{R}, i \in \mathbf{F}, k \in \mathbf{K}, s \in \mathbf{S_r}, d \in \mathbf{D} \tag{6f}$$

$$x_i^k \in \{0,1\}, \quad \forall i \in \mathbf{F}, k \in \mathbf{K} \tag{6g}$$

$$y_{ri}^{ksd} \in \{0,1\}, \quad \forall r \in \mathbf{R}, i \in \mathbf{F}, k \in \mathbf{K}, s \in \mathbf{S_r}, d \in \mathbf{D} \tag{6h}$$

$$z_{rij}^{kmsd} \in \{0,1\}, \quad \forall r \in \mathbf{R}, i \in \mathbf{F}, k \in \mathbf{K}, s \in \mathbf{S_r}, d \in \mathbf{D} \tag{6i}$$

where (6d)-(6f) are binding constraints that insure z_{rij}^{kmsd} taking the same value as product $y_{ri}^{ksd} y_{rj}^{msd}$.

IV. A Scale Free Heuristic Approach

The PCC problem falls within the family of $\mathcal{NP}$-hard problems since it resemples a binpacking problem and as a result heuristics, becomes the only viable option of finding competitive feasible solutions for real time operation. Therefore, a heuristics algorithm named, Probability-prior proactive caching-chaining (P-PCC) is proposed and is detailed in the pseudocode Algorithm I below. The main philosophy of the proposed P-PCC heuristic is to create a set of candidate proactive caching points for each possible visited access router and then weighted by the probability of visiting each access router and explore node combinations for creating the service chain.

1) For any request r, select the target node $d \in \mathbf{D}$ by highest ρ_d and find the closest starting node $s \in \mathbf{S_r}$ by minimum shortest path routing cost P_{sd};
2) On the shortest path from the selected s and d, find all candidate nodes by $\mathbf{K}$;
3) Choose the closest k from the selected s on the path to host the NF$_i$ with the lowest visiting order sequence in request r if there are enough resources to support the function, otherwise, host the sub-lowest function, until running out of resources;
4) Repeat step 2 and 3 until all NFs of request r are hosted.

V. Numerical Investigations

In this section, we provide a wide set of numerical investigations to evaluate the performance of proactive chaining-caching problem under various network scenarios. The applied random networks are composed by an range of 10 to 20 candidate VNF hosting nodes and each candidate node has its degree ranged from 2 to 5. Besides, the number of starting points and destination points are set from 1 to 5. We assume that, the number of requests to be supported is 20, and the number of different VNFs is randomly selected from 1 to 3.

The moving probability to each destination node is randomly generated between 0 and 1, notice that, the summation

Algorithm 1: P-PCC

Input : **G**; **D**; **R**; **K**; **F**; **H**; attaching node o;
Output: VNF allocation: x_i^k; P-PCC cost: P-PCC;
P-PCC← 0;
for $k \in \mathbf{K}$ **do**
  Remaining utility of node k: $RU_k \leftarrow U_k$;
end
:// VNF Allocation
for $i \in \mathbf{D}$ **do**
  if $\rho_i == max(\rho_i)$ **then**
    Destination node: $d \leftarrow i$;
  end
end
for $r \in \mathbf{R}$ **do**
  Starting node:$s \leftarrow$ find closest node s to d in S_r with minimum P_{sd};
  candidate node priority list: $CPL \leftarrow \emptyset$;
  $CPL \leftarrow$ sort $k \in \left\{ \{n|n \text{ is on the shortest path from s to d}\} \cap \mathbf{K} \right\}$ by the distance between k and s from low to high;
  VNF priority list: $FPL \leftarrow \emptyset$;
  $FPL \leftarrow$ sort f_i by its visiting sequence l of r;
  for $k \in CPL$ **do**
    for $f_i \in FPL$ **do**
      if $u_i \leq RU_k$ *and* f_i *is not hosted at* k **then**
        host f_i at k ;
        $x_i^k \leftarrow 1$;
        label f_i as hosted at k ;
        $RU_k \leftarrow RU_k - u_i$;
        P-PCC ← P-PCC+C_i^k;
      end
    end
  end
end
x_i^k :// Chaining Routing cost
for $d \in \mathbf{D}$ **do**
  for $r \in \mathbf{R}$ **do**
    Length of the chain requested by r: $L_r \leftarrow$ the number of requested VNFs by r;
    VNF chaining list: $I \leftarrow$ sort f_i by its visiting sequence l of r;
    Chaining Routing cost between $I(j)$ and $I(j+1)$: $CR_{j,j+1} \leftarrow$ cumulative P_{km} where k hosts $I(j)$ and m host $I(j+1)$ which is given by x_i^k;
    Chaining Routing Cost: $CRC \leftarrow 0$;
    $CRC \leftarrow$ find the cost of shortest chaining routing path from $I(0)$ to d;
    P-PCC← P-PCC+$\rho_d CRC$;
  end
end
P-PCC;

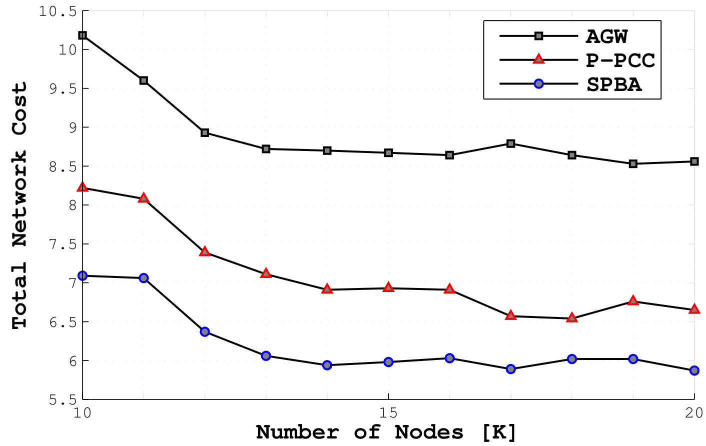

Fig. 5: Performance of the proposed scheme with different number of nodes in the network.

of the moving probability to each destination of a mobile user does not exceed 1. The shortest path routing cost and the VNF placement cost are measured by general network metrics in the open interval of (0, 100]. In terms of physical resources of candidate VNF hosting node, we assume that each candidate node has 2 GByte memory capacity and 16 virtual CPU cores. While each VNF consumes memory in a range from 10 to 50 MByte and use 0.5 to 1 cores. The results are obtained by averaging 1000 Monte Carlo simulations. To sum up, the parameters that have been used in the investigations are presented in Table I.

TABLE I: Simulation Parameters

Parameter	value
Number of candidate hosting nodes(K)	10-20
Degree per node	2-5
Moving probability (ρ_d)	0-1
Number of starting points per request ($\mathbf{S_r}$)	1-5
Number of destination points per user ($\mathbf{D}$)	1-5
Number of requests to support (R)	20
VNF number (F)	1-3
Maximum number of VNF in a chain (L)	1-3
Cost metric per link (P_{km})	(0,100]
Cost metric per node to host VNF(C_i^k)	(0,100]
Memory capacity per candidate node	2 GByte
Number of virtual CPU cores per candidate node	16
Memory requirement per VNF	10-50 MByte
CPU core requirement per VNF	0.5-1

The proposed scheme is compared with two baseline schemes. In the first one, which provide a lower bound on the performance, content caching and VNFs are hosted at the network gateway, namely, AGW. The second scheme allocates caching as well as VNFs along the shortest path from the gateway node to the serving access router without considering mobility, and is called Shortest Path Based Allocation (SPBA). Figure 5 shows the performance of the proposed scheme compared to the previous mentioned baseline techniques for different number of nodes in the network. As can be seen from the figure a significant performance gain of over 12% can be achieved which is robust against different network

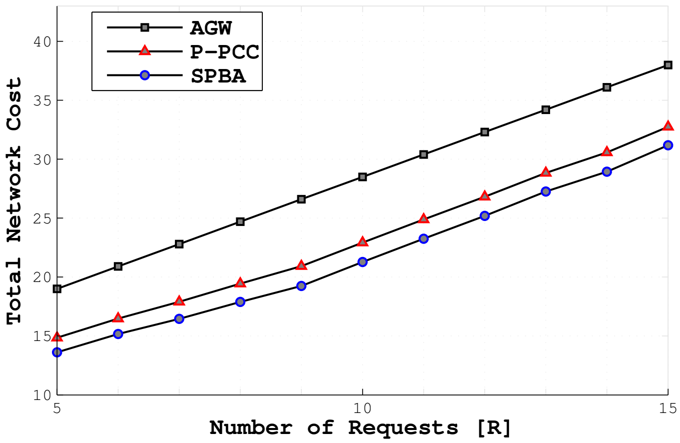

Fig. 6: Performance of the proposed scheme with increased number of service requests in the network.

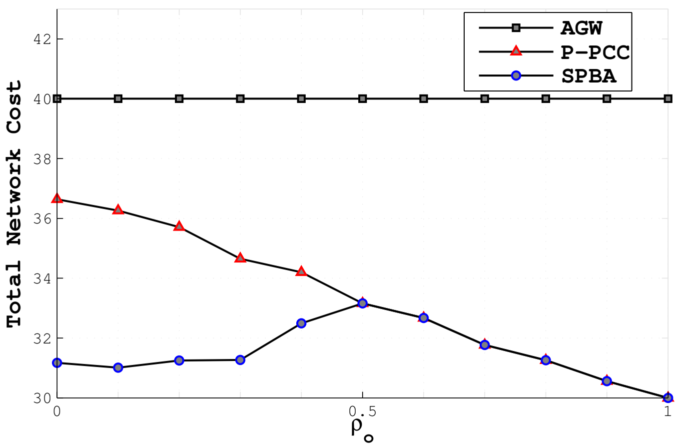

Fig. 7: Performance of the proposed scheme for different values of the parameter ρ_0.

sizes. A similar observation can be made from figure 6, which shows the performance for different number of requests. With increased number of requests, i.e., more constrained allocations, the performance gains increase from 4.8% to 8.3%. Finally, in figure 7 we show the performance of the proposed scheme for different mobility use cases. The figure shows the performance gains as a factor of the parameter ρ_o. This parameter is defined as follows $\rho_o = 1 - \sum_{d \in \mathbf{D}} \rho_d$, which means that as ρ_o reaches close to 1 there is no mobility of the end-user, i.e., there is no change on the serving access router. As expected, there are no gains when there is no mobility, but as the mobility increase the gains reach up to 15%.

VI. Conclusions

In this paper, the rational of joint proactive caching and VNF chaining has been presented together with some key observations on this problem and the general principle of optimizing cache specific VNF service chains. Based on those preliminaries an optimization framework using integer linear mathematical programming has been detailed that integrates VNF chaining with proactive caching. In addition, since the problem resembles the multi-dimensional binpacking problem, which is NP-hard, a scale-free heuristic algorithm has been presented that can be applied in large network instances amenable for real time implementations. Finally, the attainable performance of the proposed proactive caching service chains schemes was investigated.

Acknowledgment

This work has been performed in the framework of H2020-ICT-2014-2 project 5G NORMA. The authors would like to acknowledge the contributions of their colleagues, although the views expressed are those of the authors and do not necessarily represent the project.

References

[1] R. Mijumbi et al., “Network function virtualization: State-of-the-art and research challenges,” *IEEE Commun. Surveys Tuts*., Vol. 18, No. 1, 2016
[2] Network Function Virtualisation: ETSI introductory white paper, http://portal.etsi.org/NFV/NFV_White_Paper.pdf, October 2012.
[3] Gember A., Krishnamurthy A., Saul St. John, Grandl R, Gao X., Anand A, Benson T., Sekar V., Akella A. (2013), Stratos: A Network-Aware Orchestration Layer for Virtual Middleboxes in Clouds, Technical Report, 2013
[4] Rajagopalan S., Williams D., Jamjoom H., Warfield A., Split/Merge: System Support for Elastic Execution in Virtual Middleboxes, ACM USENIX Symposium on Networked Systems Design and Implementation, 2013
[5] Riera F., et al., On the complex scheduling formulation of virtual network functions over optical networks, 16th International Conference on Transparent Optical Networks (ICTON), 2014
[6] T. Han and N. Ansari, Opport unistic content pushing via WiFi hotspots, in Proc. 3rd IEEE IC-NIDC, Sep. 2012, pp. 680âĂŞ684
[7] K. Kanai et al., Proactive Content Caching for Mobile Video Utilizing Transportation Systems and Evaluation Through Field Experiments, in IEEE Journal on Selected Areas in Communications, vol. 34, no. 8, pp. 2102-2114, Aug. 2016.
[8] R. Guerzoni, “Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges and Call for Action. Introductory white paper”, in *Proc. SDN and OpenFlow World Congress*, Jun. 2012 pp. 1-16.
[9] S. Lee, S. Pack, M-K. Shin, E. Paik, R. Browne, Resource Management in Service Chaining, Internet Research Task Force (IRTF), Internet-Draft, July 2015
[10] X. Jin, L. Erran Li, L. Vanbever and J. Rexford, “SoftCell: Scalable and flexible cellular core network architecture”, *Proc. 9th Int. Conf. Emerging Netw. Exp. Technol*., pp. 163-174, 2013.
[11] O. N. Fundation, “Software-defined networking: The new norm for networks,” *ONF White Paper*, 2012.
[12] “ETSI GS NFV 002 V1.2.1: Network Functions Virtualisation (NFV); Architectural framework,” ETSI Ind. Spec. Group (ISG) Netw. Functions Virtualisation (NFV), Sophia-Antipolis Cedex, France, Dec. 2014. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf
[13] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, and G. Polyzos, “A Survey of Information-Centric Networking Research”, in *IEEE Commun. Surv. Tut*. (available online since 19 July 2013).
[14] X. Wang, “Cache in the Air: Exploiting Content Caching and Delivery Techniques for 5G Systems”, *IEEE Commun. Mag*., vol. 52, no. 2, pp. 131-39.
[15] G. Zheng, V. Friderikos, “Optimal proactive caching management in mobile networks” in *Proc. IEEE ICC*, 2016.
[16] V. A. Siris, X. Vasilakos, and G. C. Polyzos, “Efficient proactive caching for supporting seamless mobility,” *arXiv preprint arXiv*:1404.4754, 2014.
[17] S. Mehraghdam, M. Keller, and H. Karl, “Specifying and placing chains of virtual network functions,” in *Proc. IEEE 3rd Int. Conf. CloudNet*, Oct. 2014, pp. 7-13.
[18] “ETSI GS NFV 001 V1.1.1: Network Functions Virtualisation (NFV); Use Cases,” ETSI Ind. Spec. Group (ISG) Netw. Functions Virtualisation (NFV), Sophia-Antipolis Cedex, France, Oct. 2013. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/nfv/001_099/001/01.01.01_60/gs_nfv001v010101p.pdf