Experimenting with cache peering in multi-tenant 5G networks

Konstantinos V. Katsaros, Vasilis Glykantzis
Intracom SA Telecom Solutions,
Email:{konkat, vasgl}@intracom-telecom.com

***Abstract*—The introduction of virtualized compute and storage resources at the edge of 5G mobile networks is expected to facilitate the flexible instantiation and management of services and applications at the network vicinity of the end users, including content caching. At the same time, virtualization fosters the emergence of multiple-tenants of the shared 5G infrastructure, separately operating such services. In this context, this paper examines the potential benefits presented by the cooperation of co-located tenants in the form of cache peering relationships. Building a proof of concept testbed, we experimentally validate these benefits further investigating aspects related to resource allocation and isolation in a sliced networking environment. Our preliminary results show that cache peering can increase cache hit ratio by approximately 15.3%, an increase otherwise achieved by the lease of an additional 33% of storage resources. This comes at the cost of limited CPU utilization overheads (8.7%) due to peering load.**

***Index Terms*—5G, NFV, SDN, MEC, multi-tenancy, caching**

I. Introduction

The introduction of virtualized compute, storage and network resources in 5G mobile networks, as dictated by the emerging Network Functions Virtualization (NFV) [1], but also Mobile/Multi-access Edge Computing (MEC) [2] paradigms, promises a series of advantages related to elasticity and flexible lifecycle management of network services, but also low-latency performance when application/service functionality is placed in the network proximity of users. The virtualized nature of the resources facilitates the sharing of the physical infrastructure by different actors *i.e.*, *tenants*, promoting competition and service differentiation. This facilitates the emergence of Virtual Network Operators (VNOs), paving the way for mobile (edge) networks shared by multiple tenants, each one realizing its own network functions and services towards its customer base.

Among these functions and services, caching is expected to play an important role. Indeed, the high volumes of content delivery traffic observed in today's mobile networks and the projections for a rapid increase in the foreseeable future [3], are expected to put (virtual) network infrastructure under severe stress. Caching is already employed as an important countermeasure, typically in the form of in-network cache middleboxes caching content traversing the network, transparently to content providers [4]. Non-transparent (to content providers) caching schemes further enable network operators enter the content delivery network (CDN) market, offering their services to customer content providers. In this technological and business context, the advent of the aforementioned virtualization capabilities is only expected to foster the adoption of caching solutions, implemented as virtual network functions (VNFs), also by VNOs.

However, the shared nature of the 5G mobile (edge) infrastructure and the emerging multi-tenancy environments, present new opportunities and challenges for caching services. Overlooking the potential co-location of VNO caches on the same physical infrastructure can potentially result in inefficient network and server resource utilization *i.e.*, virtual caches (vCaches) may cache the same content while residing on the same (micro-) data center (μDC) / MEC server. Instead, as we discussed in our previous work [5], the co-location of VNOs bears the potential for synergies to remedy this situation, further paving the way for realization of new business opportunities in 5G. Cooperation can take the form of cache peering agreements, in which VNOs co-located on the same μDCs mutually agree to exchange content already cached, to avoid redundant communication with the origin content server, while also reducing redundancy in cache storage utilization (see Figure 1).

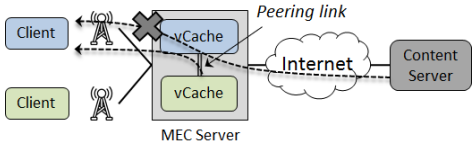


Figure 1: vCache peering concept

In this paper, we take a closer look at the envisioned cache peering relationships in multi-tenant 5G networks, with the purpose to quantify the potential performance benefits, but also shed light on the associated resource utilization overheads, in the context of multi-tenancy and resource isolation. Based on a prototype implementation, we engage in measurements that quantify trade-offs related to the symmetry of the load and the allocated resources in the peering domains. Taking into account the developments on the MEC paradigm, we focus this work on edge network deployments. Our investigation provides initial insights for the design and configuration of the envisioned service, shedding light on aspects that can shape the corresponding business relationships. Our preliminary key findings are summarized as follows:

- We first validate and further quantify the expected cache peering benefits, showing that the proposed scheme can improve cache hit ratio and download times by 15.3% and 2.55% respectively. Similar improvements can be achieved without peering, but at the cost of an additional

33% of storage resources, motivating the orchestration of peering relationships as a cost effective means to improve cache performance.
- Cache peering has a limited impact on CPU utilization in edge environments, facilitating resource isolation between tenants. Symmetric workloads lead to an increase of 8.72%, which remains modest for heavily unbalanced workloads *e.g.*, a 400% increase on the request rate of a peering VNO results only in a 15.8% increase of local CPU utilization.

In the following, we first provide some background on cache peering and multi-tenancy, and we then present the proposed vCache peering scheme (Section II-A). We describe our performance evaluation framework and discuss our experimental our experiments (Section III). Finally, we discuss related work in Section IV and conclude (Section V).

II. Motivation and Scope

A. Background

1) Caching and cache peering: Content caching has been extensively employed to reduce network traffic, content server overheads, and download times, for more than two decades [6]. Nowadays, content caching typically takes the form of either opportunistic caching or pre-fetching. In the former case, caching is employed by network operators as a means to reduce their network traffic and improve performance for their subscribers [4]. Content is cached transparently to content providers[1], on the basis of content popularity. In contrast, pre-fetching mechanisms have been extensively used in the case of CDNs, as a service aiming to improve delivery performance for the particular content.

In both cases, cooperative caching schemes allow caches to exchange content [8], typically employing *parent-child* and/or *sibling/peering* relationships. In parent-child relationships, children forward to their parents requests they cannot locally serve (*i.e.*, cache misses). In sibling/peering relationships, a cache miss results in a potential request of the requested item from peering caches. Whether a request is actually issued depends on content availability information exchanged by the peering caches. This happens either pro-actively, based on a periodic exchange of cache contents summaries (*e.g.*, [9]), or reactively with an explicit request upon a local cache miss (*e.g.*, [10]). In contrast to a normal content request on a cache, a (reactive) request towards a peering/sibling cache does not trigger the communication of the latter with the content server upon a cache miss. As a result, the existence of a peering link does not affect the contents of a peering cache.

2) Multi-tenancy in mobile edge networks: The introduction of virtualized resources in 5G networks, fosters the sharing of the physical infrastructure by multiple actors *i.e.*, tenants. The exact nature of multi-tenancy can vary, subject to the corresponding nature and network footprint of the virtualized functions/applications/services. Multi-tenancy can take the shape of co-located content/service/application providers, Over-The-Top players, but also CDNs providing caching services to content providers. At the same time, end-to-end network services can be established for the realization of VNOs, providing generic network access to their subscribers, potentially along with other in-network services/applications. In this case, the network is divided in segments dedicated to their owners (tenants). Virtualized resources can be employed for the realization of either opportunistic/transparent caching or CDN-like services; a typical example of the NFV paradigm [1], [11].

In either context, multi-tenancy is underpinned by the principles of traffic and performance/resource isolation. Traffic is not allowed to cross network segments of tenants, enhancing security and allowing tenants to apply their individual policies. In the context of 5G mobile networks, cross-tenant communication is typically envisioned through the (virtual) network packet gateways [5]. At the same time, performance/resource isolation provides tenants with guarantees on the availability of the allocated resources, ensuring that the no tenant is allowed to consume resources of another tenant, affecting this way service performance.

B. vCache peering

In view of the mobile network evolution, our work proposes and investigates the establishment of cache peering relationships between tenants of the same mobile (edge) infrastructure[2]. Our work initially focuses on the case of opportunistic HTTP caching services deployed by co-located VNOs. The envisioned VNO synergy builds on the shared nature of the underlying resources. Figure 2 illustrates a baseline setup. The realization of each individual VNO (local) cache service utilizes management and orchestration services of the widely adopted ETSI NFV architecture [12]. In the simplest case, such services are provisioned to VNOs by the 5G infrastructure operator. VNOs instantiate vCaches as VNFs on top of the virtualized (edge) infrastructure *i.e.*, μDC / MEC server. The potential co-location of vCache instances within the same μDC, then promises a particularly low latency and high bandwidth network path between the corresponding virtual machines (VMs)/VNFs. The availability of such a communication channel between vCaches[3] is expected to facilitate and motivate the exchange of already cached content over a peering/sibling link, *i.e.*, bringing content potentially cached in a collocated vCache is expected to reduce the traffic overheads towards the content origin(s), yielding, at the same time, lower latencies for the end users, as opposed to contacting the content server. Enabling the exchange of cached content under a peering agreement, in an analogy to inter-domain traffic peering agreements (*e.g.*, [13]), further promises economic incentives related to the reduction of transit traffic costs.

However, peering goes beyond the mere exchange of traffic as the support of cache peering now involves computing and

[1]Standardization activities have already engaged in the establishment of mechanisms overcoming traffic encryption (HTTPS) and intellectual property rights (IPR) considerations [7].

[2]For simplicity, we consider only scenarios involving bilateral relationships.

[3]Section III provides technical details on how this channel can be realized in OpenStack deployments.

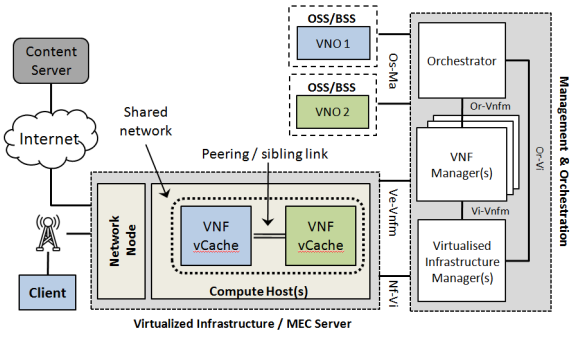


Figure 2: Baseline vCache peering setup.

storage resources utilized for cache index lookups and the potential delivery of content. It follows that the potential performance gains come at the cost of this resource utilization, which in turn may translate to either local performance degradation, going against performance isolation (see Section II-A2), and/or potentially higher operational expenditure (OpEx) for the lease of additional resources dedicated to the peering relationship [5].

Getting then a better understanding of the potential incentives for VNOs to engage in cache peering agreements goes through cost-benefit analysis aiming at a qualitative and quantitative investigation of the aforementioned aspects. In this work, we initiate such an analysis focusing on performance benefits, both for the VNOs, in the form of traffic savings (*i.e.*, expressed in cache hit ratios) and the end users[4] (*i.e.*, expressed in terms of download times), as well as costs, associated with resource utilization for the support of the vCache peering relationship (*i.e.*, expressed in terms of CPU utilization).

Such an analysis requires the careful investigation of aspects related to the encountered workload and the allocation of resources, on each side of a peering link. Increased cache peering benefits for a VNO are associated with the availability of the desired content at the other side of a peering link. However, as caching is assumed already in operation locally *i.e.*, within the VNO, cache hits on the peering link refer to content not currently cached at a VNO, but cached at its peer (termed hereby as *residual* content). Increased cache hits on the peering link relate then to the volume of cached content at both sides of a peering link, determined by the allocated cache storage space (C)[5]. At the same time, increased cache hit rates on the peering link also point to increased CPU utilization at the peering cache, consuming the corresponding virtualized resources. Further considering that cache sizes and request rates may not be identical for each VNO, it becomes apparent that the assessment of the potential cache peering benefits, along with the associated resource utilization overheads, becomes a non-trivial task. In view of this complexity, in this paper we devise a performance evaluation framework to support the experimental evaluation of the aforementioned aspects.

III. Evaluation

A. Framework

Testbed. We evaluate our baseline vCache peering scheme (see Figure 2) in a simplified testbed based on a typical OpenStack[6] deployment. For each tenant, a client component is created responsible for generating aggregate HTTP request streams towards a content server (see next). Traffic passes through the instantiated vCache via proxy configuration on the clients. Alternative approaches exist for the transparent (to the client) interception of traffic by the vCaches[7]. We expect this simplification not to substantially affect the targeted results. Traffic isolation of tenants is achieved through VLAN segments. Following the approach in [5], we enable communication between the peering caches through a *shared* network. Access rights to this network are managed by the one of the VNOs, through the Role-Based Access Control (RBAC) feature introduced in OpenStack Liberty[8].

On the application layer, we employ the Squid cache implementation[9], a mature and widely adopted solution. The peering link between the vCaches is configured through the `cache_peer` directive[10], with reactively exchanged information on content availability (see Section II-A1).We configure Squid with the default Least Recently Used (LRU) cache replacement policy. Finally, a web server is instantiated within a separate VM outside the VLAN segments of both VNOs. As web servers cannot be expected to typically reside in close network proximity to the virtualized infrastructure, we artificially introduce additional latency between both VNOs and the web server VM, following a normal distribution with a mean value of 150ms and a standard deviation of 50ms [14].

Metrics. As discussed in Section II, our investigation focuses on the average download times (ADT) and cache hit ratios (CHR) observed. We refine the assessment of the CHR, considering: (i) the local CHR (CHR_L) measured as the ratio of HTTP requests sent within a VNO that were served with content cached locally, and (ii) the CHR over the peering link (CHR_P) measured as the ratio of the client requests that were served by the peering cache. Resource consumption is captured by the CPU utilization metric reported by Squid.

Workload. Generated by the Globetraff tool [15], the workload in our experiments consists of a typical content catalog that follows a Zipf-like popularity distribution (with slope a) [4], with exponentially distributed request inter-arrival times (with rate r, expressed in HTTP requests per second.). The distribution of the content items size is modeled as the

[4]Obviously, better performance for end users translates to a competitive advantage for the respective VNO.

[5]Other aspects also play an important role, including content catalog similarity and corresponding popularities at each side of a peering link. In this paper we assume similar content catalogs, leaving this investigation for subsequent work.

[6]https://www.openstack.org/

[7]http://wiki.squid-cache.org/SquidFaq/InterceptionProxy

[8]http://docs.openstack.org/liberty/networking-guide/adv-config-network-rbac.html

[9]http://www.squid-cache.org/

[10]http://www.squid-cache.org/Doc/config/cache_peer/

concatenation of the Lognormal (body) and Pareto (tail) distributions [15]. Focusing in this work on resource allocation, and utilization/isolation aspects, we set the workload at both VNOs to come from the same content catalog, randomizing however the arrival of content item requests at each VNO.

B. Results

Baseline peering scenario. Figure 3 illustrates the evolution of the observed cache hit ratio, download times and CPU utilization metrics during a baseline experiment: both VNOs present the same workload characteristics ($a = 0.8$, $r = 1req/sec$) [4] and have the same cache capacity (1% of the catalog size). As the scenario is symmetric, results only present the observed values for one of the VNOs. Figure 3(a) shows an increase of the overall cache hit ratio throughout the experiment (after the initial warm up period), which corresponds to a 15.3% increase on average (*i.e.*, from 25.24% to 29.11%). Download times are also reduced by approximately 22ms on average (approximate decrease of 2.55% on average) (Figure 3(b)). Figure 3(c) shows a consistent increase of CPU utilization in the peering scenario. On average CPU utilization is increased by 8.72% (*i.e.*, from 13.76% to 14.96% in our scenarios). This is considered as a modest penalty for the support of cache peering, tightly linked to the low aggregate request rates expected in the edge of the mobile network.We revisit the impact of high request rates later on in this section.

In all, though the 15.3% increase of CHR (and corresponding traffic savings) may not appear as substantial at first sight, it is important to assess it in the light of potential alternatives for VNOs. Typically, a targeted increase of the CHR would involve an investment on additional storage space. Figures 4(a) and 4(b) examine this case comparing cache peering benefits with those of an increase of storage space. We see that the total CHR benefits of vCache peering compare to those of an approximate increase of 33% of the leased storage space, though without necessitating the allocation of additional resources (and the associated OpEx). This does not hold for the ADT values observed, since an increase of storage space results in local cache hits, as opposed to hits on the peering cache in the case of vCache peering. However, still, peering achieves a decrease of ADT by 2.55% (*i.e.*, by 22ms on average) against a decrease by 6.5% (*i.e.*, by 55.8ms on average) in the case of a 33% cache capacity increase.

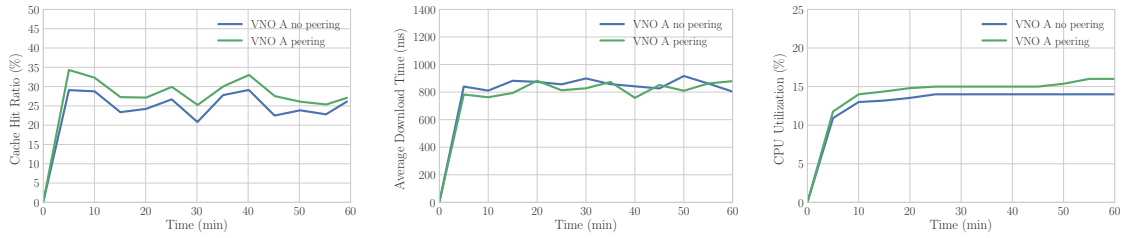
Non-symmetric cache capacity. We next investigate the effect of non-symmetric cache sizes, corresponding to scenarios where (i) VNOs do not (truthfully) coordinate on the dimensioning on their caches, or (ii) VNOs establish peering links between already existing vCaches with set capacities. Figure 4(c) shows the CHR achieved in scenarios where $C_A = 1\%$ and $C_B \in \{1\%, 2\%, 3\%, 4\%\}$. For VNO B, a substantial increase of cache capacity does not bring analogous (linearly increasing) benefits on CHR, as expected, due to the heavily skewed distribution of popularity. It is interesting through to observe a corresponding increase of CHR at the peering VNO A, with peering cache hits (CHR_P) increasingly contributing to the overall CHR as C_B increases.

Non-symmetric request rates. Figure 5 shows the effect of non symmetric request rates on CPU utilization, for $r_A = 1req/sec$ and $r_B \in \{r_A, 2r_A, 3r_A, 4r_A$. As expected, VNO B experiences a substantial increase of CPU utilization, as the local load increases. However, we see that the impact on VNO A is modest since, for instance, a 400% increase of the total request rate at VNO B yields an increase of 15.8% in CPU utilization for VNO A, but a corresponding 193.9% increase for VNO B. Taking a closer look on this result, it is first important to recall that even increased request rates on the peering link (as a consequence of the overall increase at VNO B), do not affect the contents of the cache in VNO A, thus keeping a stable local CHR (not shown due to length limitations) (see Section II-A1). As a result any additional impact on VNO A CPU utilization owes only to the increased number of ICP queries and the increased number of content transfers (hits), approximately 3.36% of the total ICP queries. Interestingly, this load increase is not sufficient to significantly affect CPU utilization at VNO A, lowering the resource consumption overheads for peering VNOs, even in the presence of heavily unbalanced loads.

IV. Related work

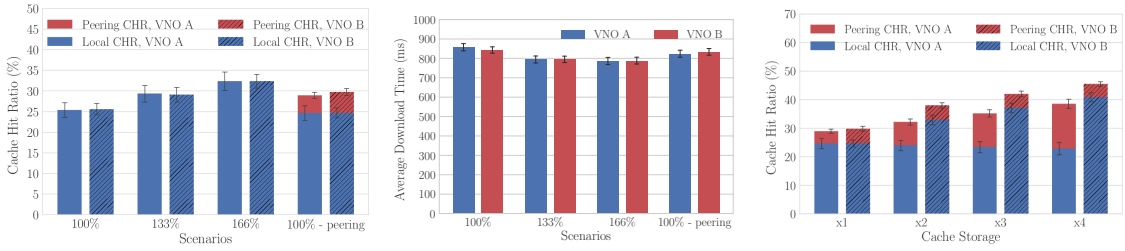
Caching has been identified as a key function, right from the beginning of the NFV concept [1]. Since then, commercial NFV-enabled solutions have appeared, including caching as a key building block of broader, CDN-oriented solutions e.g., [11]. Additionally, the currently on-going 5G PPP EU H2020 research projects put substantial effort in adopting the NFV paradigm within the 5G landscape [16]. However, to the best of our knowledge, no commercial solution or research effort focuses on the particular challenges of cache peering in multi-tenancy environments. Efforts have also been devoted to the extension of (caching) service footprint through peering, in the context of Content Delivery Network interconnection (CDNi) [17]; however, these efforts aimed at the design of interfaces between CDNs, rather than building on the emerging NFV capabilities. Beyond system design aspects, Pacifici *et al.* perform an interesting investigation on the convergence of autonomous peering caches in future content-centric networks [18]. The study shows that under imperfect information on expected content popularity, the considered opportunistic caches tend to converge into cost-efficient cache allocations, further motivating the establishment of the proposed mechanisms in more complex environments.

In all, we consider the proposed cache peering approach as a step beyond mere virtualized caching services, focusing on the opportunities brought in the field by NFV, namely, for multi-tenancy. This brings corresponding opportunities for new business models for the NFV-enabled cooperation between VNOs, in an analogy to inter-domain traffic peering agreements [13]. However, cache peering further involves the implicit sharing of compute and storage resources, calling for the identification of the related challenges and appropriate approaches. Our work here aims at taking a first step in this direction.



(a) Cache Hit Ratio (b) Download Time (c) CPU Utilization

Figure 3: Effect of vCache peering link on CHR (a), ADT (b) and CPU utilization (c), for VNO A in a symmetric scenario with identical cache capacity and workload characteristics for both peering VNOs.



(a) Cache Hit Ratio (b) Average Download Time (c) Cache Hit Ratio

Figure 4: Effect of cache capacity on CHR (a) and ADT (b). In (c), a non-symmetric cache storage allocation increases CHR on both sides of a peering link.

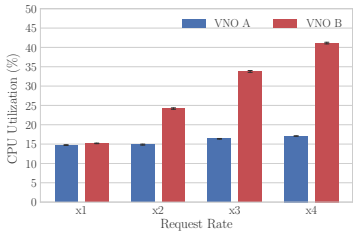


Figure 5: Effect of non-symmetric request rates on CPU utilization

V. Conclusions and Future Work

In this paper, we took the first step in evaluating the potential benefits stemming from the establishment of cache peering relationships between tenants of virtualized edge resources. Our experiments demonstrate the potential for cache hit ratio increase in the order of 15%, at the cost of a modest CPU overhead. Our next steps include broader experiments involving centralized caches and orchestration mechanisms including resource isolation design aspects.

References

[1] "Network Functions Virtualisation - Introductory White Paper," 2012. [Online]. Available: https://portal.etsi.org/nfv/nfv_white_paper.pdf

[2] M. Patel *et al.*, "Mobile-edge computing introductory technical white paper," *White Paper, Mobile-edge Computing (MEC) industry initiative*, 2014.

[3] CISCO, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 20152020," 2016. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.pdf

[4] S. Woo *et al.*, "Comparison of caching strategies in modern cellular backhaul networks," in *Proc. of ACM MobiSys*, 2013, pp. 319–332.

[5] K. V. Katsaros *et al.*, "Cache peering in multi-tenant 5G networks," in *Proc. of IFIP/IEEE 5GMan*, 2017.

[6] A. Luotonen and K. Altis, "World-wide web proxies," *Computer Networks and ISDN systems*, vol. 27, no. 2, pp. 147–154, 1994.

[7] M. Thomson *et al.*, "An architecture for secure content delegation using http," Working Draft, IETF Secretariat, Internet-Draft draft-thomson-http-scd-00, March 2016. [Online]. Available: http://www.ietf.org/internet-drafts/draft-thomson-http-scd-00.txt

[8] P. Rodriguez *et al.*, "Analysis of web caching architectures: Hierarchical and distributed caching," *IEEE/ACM Transactions on Networking (TON)*, vol. 9, no. 4, pp. 404–418, 2001.

[9] L. Fan *et al.*, "Summary cache: A scalable wide-area web cache sharing protocol," *IEEE/ACM Trans. Netw.*, vol. 8, no. 3, pp. 281–293, Jun. 2000.

[10] D. Wessels and K. Claffy, "ICP and the Squid web cache," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 3, pp. 345–357, Apr 1998.

[11] Qwilt, "NFV-based Caching and Acceleration Solution to Power Video on Mobile Networks," 2014. [Online]. Available: http://qwilt.com/qwilt-launches-nfv-caching-acceleration-solution/],

[12] ETSI, "GS NFV 002 V1.1.1, Network Functions Virtualisation (NFV); Architectural Framework," 2013.

[13] C. Labovitz *et al.*, "Internet inter-domain traffic," in *Proc. of ACM SIGCOMM*, 2010, pp. 75–86.

[14] J. Vesuna *et al.*, "Caching doesn't improve mobile web performance (much)," in *Proc. of the 2016 USENIX Conference*, Berkeley, CA, USA, 2016, pp. 159–165.

[15] K. V. Katsaros *et al.*, "Globetraff: A traffic workload generator for the performance evaluation of future internet architectures," in *Proc. of IFIP/IEEE NTMS*, May 2012, pp. 1–5.

[16] 5G PPP, "5G Infrastructure Public Private Partnership," 2017. [Online]. Available: https://5g-ppp.eu/5g-ppp-phase-1-projects/;https://5g-ppp.eu/5g-ppp-phase-2-projects/

[17] G. Bertrand *et al.*, "Use Cases for Content Delivery Network Interconnection," RFC 6770 (Informational), Internet Engineering Task Force, Nov. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6770.txt

[18] V. Pacifici and G. Dn, "Content-peering dynamics of autonomous caches in a content-centric network," in *Proc. of IEEE INFOCOM*, April 2013, pp. 1079–1087.