

Towards NFV-based Multimedia Delivery

Niels Bouten*, Jeroen Famaey[†], Rashid Mijumbi[†], Bram Naudts*, Joan Serrat[†], Steven Latré[‡], Filip De Turck*

*Department of Information Technology, Ghent University - iMinds, Gaston Crommenlaan 8/201, B-9050 Ghent, Belgium

[†]Department of Signal Theory and Communications, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain

[‡]Department of Mathematics and Computer Science, University of Antwerp, Middelheimlaan 1, B-2020 Antwerp, Belgium

Abstract—The popularity of multimedia services offered over the Internet have increased tremendously during the last decade. The technologies that are used to deliver these services are evolving at a rapidly increasing pace. However, new technologies often demand updating the dedicated hardware (e.g., transcoders) that is required to deliver the services. Currently, these updates require installing the physical building blocks at different locations across the network. These manual interventions are time-consuming and extend the Time to Market of new and improved services, reducing their monetary benefits. To alleviate the aforementioned issues, Network Function Virtualization (NFV) was introduced by decoupling the network functions from the physical hardware and by leveraging IT virtualization technology to allow running Virtual Network Functions (VNFs) on commodity hardware at datacenters across the network. In this paper, we investigate how existing service chains can be mapped onto NFV-based Service Function Chains (SFCs). Furthermore, the different alternative SFCs are explored and their impact on network and datacenter resources (e.g., bandwidth, storage) are quantified. We propose to use these findings to cost-optimally distribute datacenters across an Internet Service Provider (ISP) network.

I. INTRODUCTION

Over the past decades, the importance of multimedia services such as video streaming has increased considerably. This growth is projected to be in the range of 80% of global consumer Internet traffic by 2018 [1], causing video traffic to dominate the Internet. Several video service categories exist, ranging from classic Internet Protocol television (IPTV) services over Content Delivery Network (CDN)-based Video on Demand (VoD) HTTP Adaptive Streaming (HAS) services. Within each of these service categories, different vendors have their proprietary implementation. Although there are many similarities in functionality offered by their respective components, all of these video services require dedicated hardware to be physically deployed at different premises across the network.

In the classical network approach, functionality of a network node is strongly tied to with a physical network device. Typically, there is a dedicated network appliance for each network function (e.g., Deep Packet Inspection (DPI), CDN, Firewall). Each of these network functions has a typical location in the network (e.g., Set-Top Box (STB) at the customer edge or Quality of Experience (QoE) monitoring at the provider edge), leading to a fragmentation of non-commodity hardware across the network. As hardware life cycles are becoming shorter as technology and services innovation accelerates, the costs of physical installation and maintenance of these appliances increases. Furthermore, deploying new services over the network requires installing new configurations at different sites in the network, inhibiting a fast Time to Market of new and innovative services. Network Function Virtualization (NFV)¹

was introduced to alleviate the aforementioned problems by decoupling the network functions from physical hardware and by leveraging IT virtualization technology to allow Virtual Network Functions (VNFs) to run on standard high volume servers, switches and storage equipment.

Service deployments are often tightly coupled to the underlying network topology, imposing constraints on the service delivery. The strict ordering of network functions in delivery chains and their physical interconnections lead to increasing configuration complexity when deploying new services and network functions. Moreover, due to the physical topology, all traffic on a particular segment traverses all network functions, wether these packets require handling by the service or not. To alleviate the aforementioned issues, Service Function Chains (SFCs)² were proposed, creating a service specific overlay topology, independent of the underlying physical network. Classification is used to select both traffic entering an SFC and to alter the sequence of network functions within an SFC. Software Defined Networking (SDN) can be viewed as an enabler to realise a combination of VNFs into SFCs.

The concepts of NFV and SFC open up new business opportunities in the form of Virtual Network Function Infrastructure Providers (VNFINPs), acting as brokers between Infrastructure Providers (InPs) and Service Providers (SPs). These VNFINPs leasing the virtualized infrastructure offered by different InPs and deploying, orchestrating and interconnecting VNFs to create SFCs, that are used by SPs to offer value-added services to users. InPs can profit by replacing dedicated hardware with generic hardware, and maximize resource utilization and optimize energy usage by offering this virtualized infrastructure to remote parties. They are responsible for embedding Virtual Network (VN) requests from VNFINPs within their respective management domain. The VNFINP is responsible to deploy SFCs across different InP domains to create an end-to-end SFC. The VNFINP can offer multi-tenancy of a single VNF instance, provide multiple VNF instances for a single service to increase availability and performance or offer a pool of VNFs to a set of SPs. SPs benefit from the proposed model since it allows rapid deployment and testing in a realtime network environment, thus leveraging faster Time to Market of new services. The services benefit from the dynamic nature of the network, computing and storage resources offered by the VN, which allows them to scale dynamically based on service requirements and user mobility.

II. RELATED WORK

ClickOS aims to address the issues of physical middleboxes in the network by proposing a virtualized software middlebox

¹ETSI - NFV - <http://www.etsi.org/technologies-clusters/technologies/nfv>

²IETF - SFC - <http://datatracker.ietf.org/wg/sfc/documents/>

platform [2]. This allows SPs to implement VNFs on top of the middleware and rapidly test and deploy NFV-based SFCs.

Thouin et al. propose network cost models for VoD services based on hit ratio and size of caches [3], [4]. They focus on the storage devices but do not take the underlying network and the location of these devices into account. This work was extended by *Pandey et al.* to incorporate server waiting time, one-way delay and access network bandwidth consumption [5]. We extend their proposed approach to also incorporate caching at multiple sites and take other considerations, such as cost-optimal datacenter sizing and placement.

Previous studies identified the different costs involved with deploying data center networks [6]. *Greenberg et al.* concluded that the cost of a cloud can be broken down into costs of servers (45%), infrastructure (25%), power draw (15%) and network costs (15%). They also argue that geo-diversity allows to lower latency to the users and increases reliability, but come at an additional cost if locations are not selected carefully. *Goiri et al.* propose a tool to select optimal datacenter locations based on multiple location-dependent costs (e.g., land, electricity) [7]. We also take location-dependent costs and delay constraints into consideration and additionally optimize the deployment locations to support NFV-based multimedia services.

An example of a CDN-based video delivery service is Netflix's OpenConnect program³, designed to offload backhaul traffic for Internet Service Providers (ISPs) and increase QoE for the end-users. This requires ISPs to embed dedicated OpenConnect caching appliances directly in their network and the deployment OpenConnect peering points at Internet Exchanges (IXs). Subscribing to the OpenConnect program takes up several weeks and requires analyzing current traffic to forecasting future requirements, configuring and pre-loading the devices, shipping them to the ISP and finally racking and installing the appliances. Other CDNs use a similar architecture, where dedicated caching equipment is installed on ISP premises. Scaling and mirroring of this remote infrastructure is costly and time-consuming. Furthermore, due to the difficulties of predicting future capacity requirements, business is harmed when resources are underprovisioned, while money is wasted when overprovisioned infrastructure remains unused. NFV proposes therefore to install virtual counterparts of the dedicated appliances onto generic high performance hardware, managed and maintained by the InP, allowing not only dynamic scaling to future demand but also resource optimization and sharing with other delivery platforms.

III. SERVICE FUNCTION CHAINS FOR MULTIMEDIA DELIVERY

For each multimedia service, several SFCs exist that are able to satisfy the requirements of the service. They use different VNFs, multiple distributed VNFs or chain the VNFs in a different way. A CDN-based delivery service consists of several building blocks distributed across the delivery network. At the origin, the content is located in *Content Storage* of the SP, offering the video catalogue. The *Streaming Server* is responsible for indexing this storage, encoding the video in the required formats and providing Digital Rights Management (DRM) protection to the content. Dynamic content is mostly served from the origin servers, while static content is cached at the edge. Which CDN component is to serve the end-user

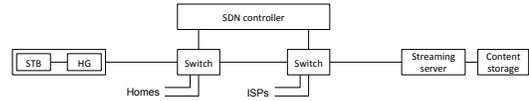


Fig. 1: High bandwidth, low storage SFC.

is decided by the *CDN Request Router*. Multiple instances of these *CDN Request Routers* are provisioned across the delivery network to minimize the impact of request delays. The *CDN Caches* are geographically distributed and located in the ISP networks closer to the edge. This limits backhaul traffic over third party networks and minimizes the delay. Other CDN components, such as QoE monitoring equipment and video transcoders could be located at the origin or closer to the edge. At the end-user premises, an *HTTP Client* or dedicated *Streaming Client* on the *Set-Top Box* can be used to retrieve the video stream. In a mobile scenario an optional *Video Transcoder* can be added to reduce the bandwidth usage on the client's mobile connection.

Figure 1 gives an example of a VoD SFC where the STB is connected directly through the Home Gateway (HG) with the *Streaming Server* without intermediate caching in the network. This SFC induces large amounts of traffic in the ISP network, but requires no additional NFV components to be deployed in the network. The delay observed at the end-user d_u is equal to the end-to-end delay $d_{o,u}$ between origin server and end-user. To reduce the network traffic, several *virtual CDN (vCDN) Caches* can be deployed within the network in combination with one or multiple *vCDN Request Routers* to coordinate the requests. Deploying additional *vCDN Caches* not only reduces bandwidth usage but also reduces content fetch delay when the content is cached closer to the end-user.

In this paragraph, we will quantify the impact of deploying caches between server and clients on the consumed bandwidth and end-to-end delay observed by the end-user as shown in the SFC in Figure 2. The impact of adding a cache on the path between client and server is dependent on the hit ratio of that cache. Suppose a cache c has a hit ratio hr_c , the amount of traffic between server and cache will be reduced with a factor of $1 - hr_c$, while the amount of traffic between the cache and the end-users will remain the same. The average delay observed at the end-user is now a weighted sum of the delay between origin server and cache $d_{o,c}$ and the delay between cache and end-user $d_{c,u}$: $d_u = (1 - hr_c) * (d_{o,c} + d_{c,u}) + hr_c * d_{c,u}$. The hit ratio hr_c depends on the portion of the total catalog with size N that can be stored in the cache c with size S_c . *Bellante et al.* report that based on Netflix traces using an oracle operating a weekly proactive prefetching, the average cache hit ratio hr_c of 27.5% for a catalog ratio of $S_c/N = 0.6\%$ (top 100 videos) [8]. Extending this to the top 1000 movies ($S_c/N = 6\%$) allows improving hr_c to 77.7%. Considering H.264 1080p HD movies of 90 minutes length encoded at $5000kbps$, the storage required for 1 movie is about $3.375GB$. The total catalog storage requirement N is thus $56.25TB$. Keeping the top 100 videos stored in cache, would require cache sizes S_c of $337.5GB$. Since the catalog size N continuously increases over time, for constant cache size S_c , the average hit ratio decreases. This pleads for a dynamic virtualized CDN infrastructure that allows reconfiguring cache sizes to attain a constant performance.

Figure 3 gives an example of a mobile VoD SFC where a *virtual Transcoder* is deployed in the SFC. This *virtual*

³Netflix OpenConnect - <http://www.netflix.com/openconnect>

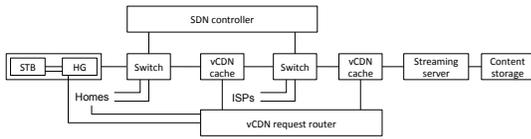


Fig. 2: Low bandwidth, high storage SFC.

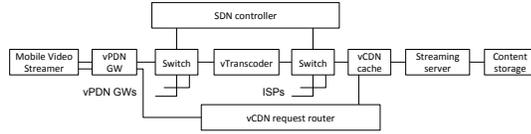


Fig. 3: Mobile user SFC.

Transcoder could either be placed in front of a cache (thus caching the transcoded content), following a cache (sharing cached content with home users) or a combination of both. Furthermore, the *virtual Transcoder* could be shared between different user groups which have similar encoding requirements. In a non-virtualized scenario, this flexibility of deploying caches and transcoders at different locations in the chain is very hard to achieve and requires manual configurations at different site locations. NFV allows alleviating these problems by supporting dynamic SFC deployment across the network.

IV. OPTIMAL DATACENTER PLACEMENT FOR MULTIMEDIA DELIVERY

A typical multi-layer network architecture consists of core, aggregation and access layers as shown in Figure 4. This hierarchical structure of Points of Presence (PoPs) has important consequences for the optimal placement and sizes of datacenters. For a video delivery service for example, allocating storage in the core layer datacenters allows serving a broad range of users from a limited number of storage sites, but incurs a huge amount of backhaul traffic, while placing storage at the access layer requires multiple smaller storage sites but minimizes the network load. The optimal allocation of datacenters in a hierarchical network topology thus boils down to finding the best balance between distributing the datacenters at increased infrastructure cost and centralizing the datacenters with increased load on the network links.

The costs of datacenter placement can be broken down into Capital Expenditures (CAPEX) and Operational Expenditures (OPEX). The CAPEX are costs related to acquiring or upgrading the physical assets, such as acquisition of datacenter equipment, property and buildings. The OPEX are ongoing costs that are incurred by operating the datacenter, such as electricity, maintenance and staff costs. Both the size and geographical location impact the CAPEX and OPEX of deploying a datacenter. The size of the datacenter impacts its surface, the

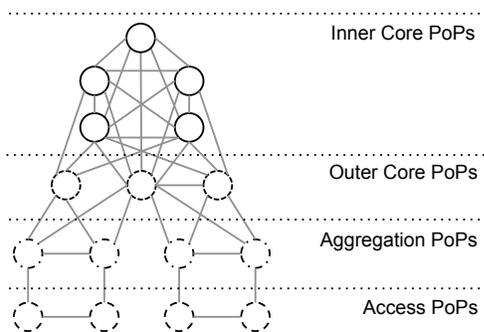


Fig. 4: Hierarchical network topology

TABLE I: Overview of possible CAPEX, OPEX and 3-year TCO for different datacenter capacities when considering server costs of 7,700\$, interest rate of 8%, PUE of 1.7, electricity cost of 0.07\$/kWh, amortization of 3 years for the servers and 12 for the datacenter

	Small	Medium	Large
Datacenter Type	Tier I	Tier III	Tier IV
Size (in <i>Watt</i>)	100,000	500,000	2,500,000
CAPEX per month (in \$/ <i>Watt</i>)	1.21	1.16	1.07
OPEX per month (in \$/ <i>Watt</i>)	0.33	0.17	0.14
3-year TCO (in \$/ <i>server</i>)	18,910	16,315	14,826

infrastructure requirements (e.g., power delivery, cooling) and network requirements (e.g., peering and connectivity links). It can be expressed as the total power (in *Watt*) that is required by a 100% utilization of the servers that will be hosted by this datacenter. The maximum power is dependent on the Power Usage Efficiency (PUE) of the datacenter that indicates the total power consumption (servers, network, cooling, etc.) relatively to the power consumption of computational equipment. A state of the art facility will typically attain a PUE of 1.7 [6].

When decreasing the size of datacenters, the CAPEX costs per *Watt* are increasing. For example, since part of the construction costs are fixed and not depending on the size of the datacenter, they can not be amortized as well as for larger datacenters. This also holds when comparing OPEX costs per *Watt*. For example, since the datacenter locations are geographically dispersed, the costs for security, maintenance, etc. increase since multiple smaller sites need to be maintained. On the other hand, since deploying multiple smaller datacenters increases the overall availability, expensive infrastructure providing redundancy such as multiple generators and UPS systems can be eliminated [6], [7]. Deploying a single large centralized datacenter however, requires high availability since there are no other datacenters which could serve as a failover in case of failure. This allows to use networks of Tier I datacenters to achieve the same availability as a single large Tier IV datacenter. Table I gives an overview of monthly CAPEX, OPEX and 3-year Total Cost of Ownership (TCO) for different datacenter sizes [9]–[12].

An important factor that needs to be considered when deploying multimedia services is the latency that is incurred by the network. Different applications have distinct levels of tolerance with respect to delay. A VoD application allows several seconds of network delay, but a Voice over IP (VoIP) application only allows acceptable QoE for one-way delays smaller than 150ms⁴. For interactive settings where users interact with each other or can control panning and zooming functions on the video content, the tolerable delay is less than 150ms. Offering services or storing content in datacenters closer to the end-user allows reducing the delay incurred by fetching the data through backhaul.

V. OPTIMIZATION MODEL

The goal is to optimally distribute datacenters across the network in order to optimize the deployment of a vCDN. For each node n , D_n is a binary decision variable indicating whether a datacenter should be located at n , while S_n is an integer decision variable indicating the size (in number of servers) of that datacenter. The size S_{c_n} of a cache c_n , is limited by the available storage of a single server (st) and

⁴CISCO - <http://www.ciscopress.com/articles/article.asp?p=357102>

thus by $S_n \times st$. The hit ratio hr_{c_n} of the cache c_n is related to S_{c_n} . \mathcal{L} is the set of links interconnecting the nodes, each link l has a capacity of C_l . For each node n , there is a set of downstream links \mathcal{L}_n^- , connecting n to the end-users \mathcal{U} and a set of upstream links \mathcal{L}_n^+ , connecting n to the content storage. Each link $l \in \mathcal{L}$ has an associated set of users \mathcal{U}_l that have l in their path to the content storage. For each link l , there is a node n for which $l \in \mathcal{L}_n^+$, we call this node n_l . The total amount of traffic T_l on a link l can thus be calculated as shown in Equation (1). If n_i is a node connecting user $u \in \mathcal{U}$, T_{n_i} is the bitrate of the requested video. \mathcal{L}_u is the ordered set of links that are traversed from a user u to the content storage. Each link l has an associated one-way-delay d_l . The average delay d_u experienced by a user u , will be dependent on the presence of a cache in the set of nodes \mathcal{N}_u along the path. Furthermore, since the packets needs to be processed by the virtual appliances, packet processing delays p_{n_i} at each node n_i along the path should be considered. Suppose a node $n_i \in \mathcal{N}_u$ is connected upstream with link $l_i \in \mathcal{L}_u$ and d_{n_i} is the one-way-delay observed up until node n_i as defined in Equation (3), the one-way-delay for user u can be calculated as shown in Equation (2).

$$T_l = (1 - D_{n_l} \times hr_{C_{n_l}}) \times \sum_{n_i \in \mathcal{L}_l^-} T_{n_i} \quad (1)$$

$$d_u = \sum_{n_i \in \mathcal{N}_u} (d_{n_i} + D_{n_i} \times p_{n_i}) \quad (2)$$

$$d_{n_i} = D_{n_i} \times hr_{C_{n_i}} \times \left(\sum_{k \in \mathcal{L}_{n_i}^-} d_k + d_{n_i} \right) \times \prod_{k \in \mathcal{L}_{n_i}^-} (1 - D_k \times hr_{C_k}) \quad (3)$$

The goal is to minimize the total cost of deploying datacenters at specific locations and additionally, minimize the cost of bandwidth consumption over all links as shown in Equation (4). This optimization is subject to constraints on the average delay d_u for all users, which is shown in Equation (5) and should be smaller than the tolerable delay for CDN services d_{CDN} . Furthermore, the capacity of each link l should not be exceeded as shown in Equation (6).

$$\min \sum_{n \in \mathcal{N}} D_n \times TCO(V_n, S_n) + \sum_{l \in \mathcal{L}} T_l \times Cost(l) \quad (4)$$

$$\forall u \in \mathcal{U}, \quad d_u \leq d_{CDN} \quad (5)$$

$$\forall l \in \mathcal{L}, \quad T_l \leq C_l \quad (6)$$

A heuristic approach based on genetic algorithms was implemented to solve the above optimization problem. In genetic programming, a population of solutions is maintained and during each iteration, new solutions are created by combining and mutating existing solutions. Natural selection occurs based on the solution fitness. To represent the solutions, we use a binary vector of size $|\mathcal{N}|$ indicating if a datacenter is installed at location n , the fitness function then evaluates the total cost as defined in Equation (4). Creating new solutions is achieved by using one-point crossover, where the k first elements of a solution are combined with the $|\mathcal{N}| - k$ elements at the end of another solution and vice versa. This creates two new solutions. The crossover point is chosen at random. Furthermore, mutations is applied to a solution with a probability of 10%. The results show that putting small datacenters in the aggregation nodes reduces the total cost (network and datacenter) with about 26% and the delay with 127ms compared to a pure network-based solution. When comparing to putting a few

large datacenters in the core of the network, the gain in total costs is about 24% and the delay is reduced by 25ms.

VI. CONCLUSION AND FUTURE WORK

In this paper we proposed how different multimedia services can be mapped onto NFV-based SFCs. We quantified the impact of different alternative SFCs on the network and datacenter resources. The costs of different alternative placements of datacenters were quantified based on their size and location in the network. Furthermore, we formulated an optimization problem combining the cost-optimization of the datacenter placement as well as the resource-optimization the SFCs. The proposed model assists the InPs to cost-optimally deploy datacenter infrastructure, while at the same time considering the constraints posed by multimedia services. The proposed approach also allows VNFINPs to evaluate the resource trade-offs that are to be considered when deploying alternative SFCs. Future work includes validating the proposed alternative SFCs using data acquired from ISPs.

ACKNOWLEDGMENT

Niels Bouten is funded by a Ph.D. grant of the Agency for Innovation by Science and Technology (IWT). This work was partly funded by Flamingo, a Network of Excellence project (318488) supported by the European Commission under its Seventh Framework Programme.

REFERENCES

- [1] C. V. Forecast, "Cisco Visual Networking Index: Forecast and Methodology 2013-2018," *Cisco Public Information*, June, 2014.
- [2] J. Martins, M. Ahmed, C. Raiciu, V. Olteanu, M. Honda, R. Bifulco, and F. Huici, "ClickOS and the Art of Network Function Virtualization," in *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'14, 2014, pp. 459-473.
- [3] F. Thouin, M. Coates, and D. Goodwill, "Video-on-Demand Equipment Allocation," in *Network Computing and Applications, 2006. NCA 2006. Fifth IEEE International Symposium on*, July 2006, pp. 103-110.
- [4] F. Thouin and M. Coates, "Video-on-demand server selection and placement," in *Managing Traffic Performance in Converged Networks*, ser. Lecture Notes in Computer Science, 2007, vol. 4516, pp. 18-29.
- [5] S. Pandey, Y. J. Won, J. W. Hong, and J. Strassner, "Dimensioning Internet Protocol Television Video on Demand Services," *International Journal of Network Management*, vol. 21, no. 6, pp. 455-468, 2011.
- [6] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The Cost of a Cloud: Research Problems in Data Center Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68-73, Dec. 2008.
- [7] I. Goiri, K. Le, J. Guitart, J. Torres, and R. Bianchini, "Intelligent Placement of Datacenters for Internet Services," in *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, June 2011, pp. 131-142.
- [8] W. Bellante, R. Vilaridi, and D. Rossi, "On Netflix Catalog Dynamics and Caching Performance," in *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2013 IEEE 18th International Workshop on*, Sept 2013, pp. 89-93.
- [9] W. P. Turner and J. H. Seader, "Dollars per kW plus Dollars per Square Foot Are a Better Data Center Cost Model than Dollars per Square Foot Alone," *Uptime Institute*, 2006.
- [10] W. P. Turner and K. G. Brill, "Cost model: dollars per kW plus dollars per square foot of compute floor," *Uptime Institute*, 2008.
- [11] U. Hoelzle and L. A. Barroso, *The Datacenter As a Computer: An Introduction to the Design of Warehouse-Scale Machines*, 1st ed. Morgan and Claypool Publishers, 2009.
- [12] L. A. Barroso, J. Clidaras, and U. Hlzle, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Second Edition*, 2013.