Performance Evaluation of Video Distribution in the Telecom Cloud

Lluís Gifre and Luis Velasco*
Optical Communications Group (GCO), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
**e-mail: lvelasco@ac.upc.edu*

ABSTRACT

A hierarchical cache architecture for the telecom cloud is investigated. Intermediate cache (IC) nodes, placed in geographically distributed datacenters, are interconnected through permanent “per content provider” (CP) virtual network topologies (CP-VNT), whilst leaf cache (LC) nodes are placed close to users. CP’s video contents are replicated in the IC nodes through the CP-VNTs, whilst LC-to-IC anycast connections are established periodically for content synchronization.

Topology creation and anycast provisioning problems are first formally stated and heuristic algorithms are proposed. Exhaustive simulation results show significant improvements in supported traffic. Algorithms are then experimentally validated within an implementation of the Applications-based Network Operations architecture.

Keywords: datacenter interconnection, optical networks, content distribution, in-operation planning.

1. INTRODUCTION

As current sustained traffic growth is expected to strain capacity of today’s metro network [1], novel content distribution architectures where contents are placed closer to the users are being investigated. In that regard, telecom operators are preparing their networks to the cloud [2] and deploying datacenters (DCs) in metro areas thus, reducing the impact of the traffic going from users to DCs [3]. However, effective solutions to distribute contents to those metro DCs and to synchronize contents among them need to be investigated.

In fact, live-TV distribution is in the portfolio of many telecom operators aiming at entering into competition with on-line broadcasters, such as Netflix. To this end, a Content Delivery Network (CDN) can be configured in the telecom cloud infrastructure being deployed, placing cache nodes in DCs. To reduce traffic in the core network, transcoding of live TV channels can be done in metro DCs, as close as possible to the end users [4]. In addition to live contents, leaf cache (LC) nodes in the metro area need to be fed with TV cache-up programs, series and films and therefore, connectivity from an Intermediate cache (IC), storing content provider’s (CP) video contents, to LC nodes is needed. Several IC nodes can be placed in geographically distributed DC locations and interconnected through a permanent CP virtual network topology (CP-VNT), so that contents in any IC nodes are replicated through the CP-VNT. In this paper, we evaluate the performance of different CP-VNT sizes and experimentally assess the feasibility of controlling the network architecture under study.

2. ARCHITECTURE

Figure 1 shows the hierarchical cache architecture. The architecture brings benefits, including the reduction of traffic in the core as well as its inherently high availability against IC node failures. However, as a result of considering several IC nodes, a CP-VNT topology needs to be created among them. In this paper we assume the CP-VNTs are created as a tree topology and some recovery is used in case of failure. Soft-permanent IC-to-IC (I2I) connections support virtual links connecting IC nodes; the CP-VNT allows IC nodes to be permanently synchronized, i.e. any modification performed in the contents in one IC node is propagated to the rest of the IC nodes through the CP-VNT. In contrast, contents in LC nodes are synchronized periodically setting-up LC-to-IC (L2I) connections to any of the IC nodes (anycast connections).

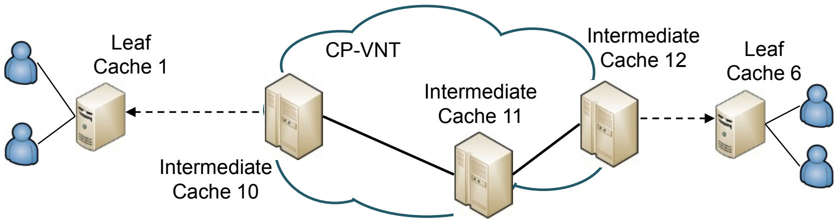


Figure 1. Hierarchical cache architecture.

Figure 2 presents an example with 3 IC nodes (labelled as 10, 11 and 12) and 2 LC nodes (1 and 6). IC nodes are connected among them by I2I virtual links supported by lightpaths set-up over the optical network (blue lines in Fig. 2a) forming a CP-VNT (Fig. 2b). When a LC node needs to download contents, a dynamic L2I connection to any of the IC nodes is set up. For example, in Fig. 2a, two L2I connections (orange lines) are established: 1-10 and 6-11.

Hence, two problems have been identified: *i*) the CP-VNT creation problem consists in finding a tree connecting IC nodes and *ii*) the L2I provisioning problem aims at finding an anycast connection between a LC node and one of the IC nodes. The next section defines and proposes algorithms for each problem.

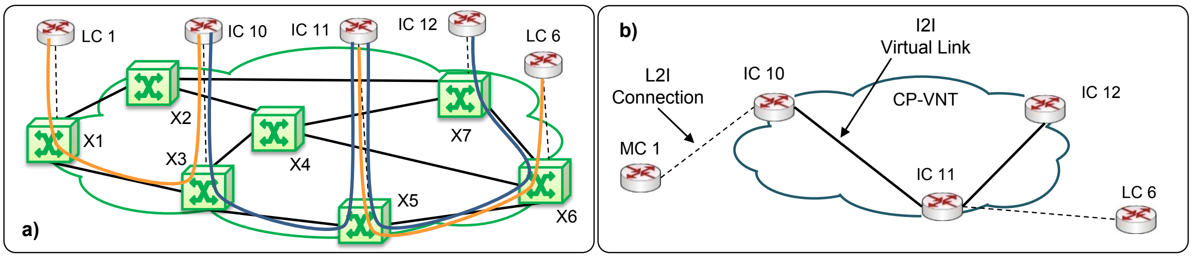


Figure 2. CP-VNT with two L2I anycast connections set up. Physical (a) and virtual (b) topologies are shown.

3. ALGORITHMS

3.1 CP-VNT_CREATE Problem

This problem consists in finding a tree connecting all IC nodes selected. Our approach consists in creating an auxiliary full-mesh graph connecting all involved IC nodes. Then, a Minimum Spanning Tree (MST) is computed over the auxiliary graph. The links in the resulting tree are the lightpaths that are set-up. To improve the availability of the CP-VNT, each optical link is used just once; therefore, a single link failure would disconnect CP-VNTs into two connected graph components at the most.

The CP-VNT_CREATE problem can be stated as follows:

Given:

- the optical network topology represented by a graph $G(N, L)$, where N is the set of optical nodes and L the set of optical links.
- the set of IC nodes, N_I.
- the number of available slices η_l in each optical link $l \in L$,
- the number of slices χ_I for the virtual links of each VNT.

Output: the route for each bidirectional lightpath.

Objective: minimize the number of optical resources required.

The algorithm is presented in Table 1. The algorithm starts creating a full-mesh graph $G(N_I, E)$ and assigning a metric to each link in E as a function of number of optical hops for the lightpath supporting that virtual link (lines 1-3). Then, a MST is computed over $G(N_I, E)$ (line 6). The links in the tree are iteratively routed over the optical network [5] (lines 9-16) and optical links already used (set L_R) are not considered for routing new lightpaths (line 10). If no path is found for a virtual link, it is removed from the full-mesh graph and the routing restarts (lines 11-13); otherwise, the path is added to the *LSP* set and sets are updated (lines 14-15). The solution in *LSP* is eventually returned (line 16).

Table 1. CP-VNT_CREATE Heuristic Algorithm

```
INPUT: G(N, L), $N_I$, $\chi_1$        OUTPUT: LSP
 1: G($N_I$, E) ← Create_Full_Mesh_Undir($N_I$)
 2: for each e=(a, b) in E do
 3:    e.metric ← |Sortest_Path(G($N_I$, E), a, b)|
 4: while TRUE do
 5:    $L_R$ ← ∅; LSP ← ∅
 6:    $E_T$ ← MST(G($N_I$, E));
 7:    if $E_T$ = ∅ then return ∅
 8:    while $E_T$ ≠ ∅ do
 9:       e=(a, b) ← argmax$_{e.metric}$($E_T$)
10:       p ← Sortest_Path(G(N, L \ $L_R$), a, b, $\chi_1$)
11:       if p = ∅ then
12:          E ← E \ {e}
13:          break
14:       LSP ← LSP ∪ {p}
15:       $L_R$ ← $L_R$ ∪ p; $E_T$ ← $E_T$ \ {e}
16:    if $E_T$ = ∅ then return LSP
```

3.2 L2I-PROV Problem

The L2I-PROV problem aims at finding an anycast connection between a selected LC node and one of the IC nodes. The problem can be stated as follows:

Given:

- the network topology represented by a graph $G(N, L)$,
- the number of available slices in each link l,
- the number of slices for the connection.

Output: the route for the lightpath.

Objective: minimize the number of optical resources required.

The approach we devised to solve this problem consists in building a graph with links connecting the LC node to all the IC nodes. As before, the metric of each link is a function of number of optical hops for the lightpath supporting that link. The graph is augmented adding one dummy node and links connecting each IC nodes to the dummy node (Fig. 3a), where the dummy node is labelled as *D*. Then, a shortest path is computed between the LC node and the dummy node. The resulting path contains the optimal route for the L2I anycast connection to be established (Fig. 3b).

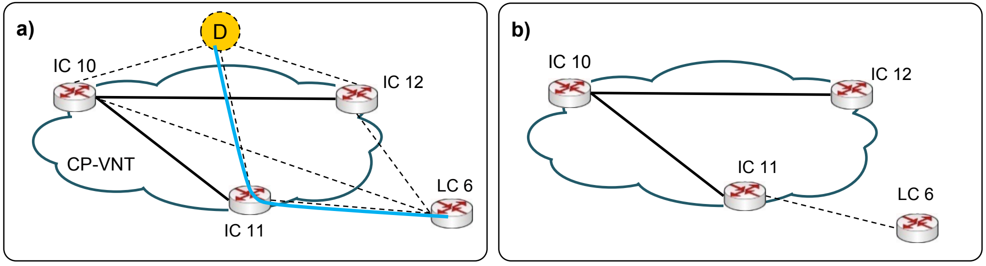


Figure 3. L2I provisioning.

4. ILLUSTRATIVE RESULTS

4.1 Performance Evaluation

The performance of the hierarchical cache architecture has been evaluated on an event-driven simulator based on OMNeT++ using the 24-node 43-link US topology. DCs were placed in every location, and so LC nodes, while up to 6 IC nodes can be strategically placed in DCs in NY, FL, IL, TX, WA, and CA. Each DC is connected to the optical network through a L2 switch equipped with 100 Gb/s transponders. A dynamic environment was simulated, where the CP-VNTs are initially created and then L2I anycast connection requests arrive following a Poisson process. The holding time of the L2I connection requests is exponentially distributed with a mean value equal to 1 h. LC nodes are uniformly chosen. Different loads are created by changing the arrival rate.

Let us first analyze the L2I provisioning performance. Figure 4a shows the blocking probability as function of the offered load. Each plot in the graph is for a fixed number of IC nodes, ranging from 1 to 5. As observed, the effects of increasing the number of IC nodes has a direct effect on the blocking probability; assuming a target 1% blocking probability, 66% more traffic can be served when CP-VNTs consist in 2 IC nodes compared to one single IC node. The rationale behind this is related to the length of the L2I connections (Fig. 4b); the number of hops to reach the IC nodes decreases when more IC nodes are considered, especially when the selected IC nodes are geographically distributed.

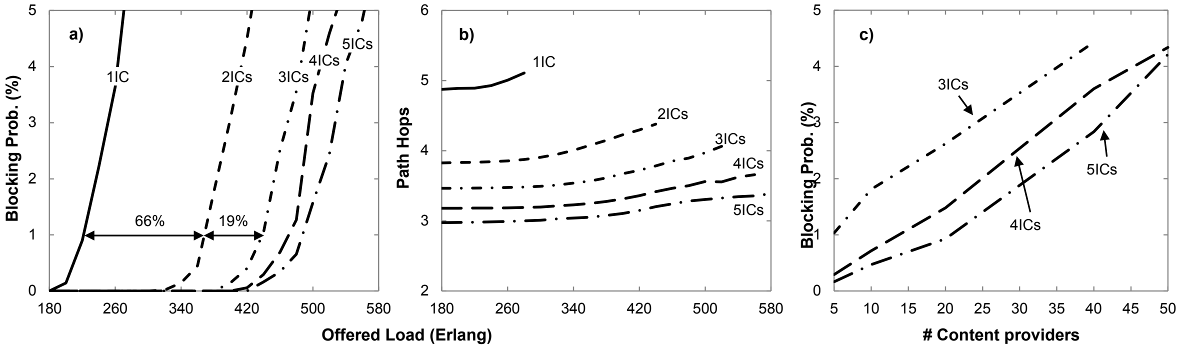


Figure 4. L2I Blocking (a) and connection hop length (b) for 5 CP-VNTs vs. load. Blocking vs. number of CPs (c).

In addition, since several broadcasters, each owning a different CP-VNT, can share the same infrastructure, let us analyze the effect of increasing the number of CPs, while keeping the total load constant. We selected the load (440 Er) for which having 3 IC nodes returns around 1% blocking (Fig. 4a). As shown in Fig. 4c, increasing the number of CPs has an almost linear effect on blocking, proportional to the number of CP-VNTs to be created.

4.2 Experimental Assessment

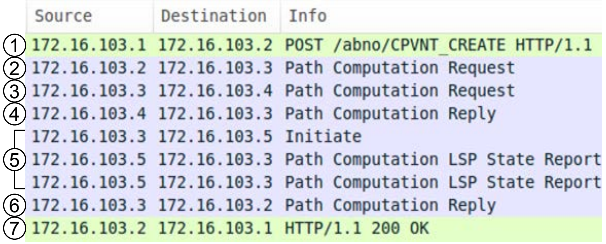
Once the performance of the proposed architecture has been studied, let us now focus on its experimental validation. Experiments were carried out in UPC's SYNERGY test-bed. An Applications Service Orchestrator (ASO) module [6] was in charge of issuing network-semantics requests to iONE [7], UPC's implementation of the Applications-based Network Operations (ABNO) architecture. ASO and all the ABNO components were implemented in C++ for Linux. ASO issues XML-encoded messages to the ABNO controller (CTRL) through its HTTP REST API, while the ABNO components exchange PCEP messages. The network shown in Fig. 2 was assumed. Control and management modules run in the IP subnetwork 172.16.103.X. Specifically, ASO runs in .1, the CTRL in .2, front-end Path Computation Element (PCE) (fPCE) in .3, back-end PCE (bPCE) in .4, and Provisioning Manager (PM) in .5. An emulated data plane was deployed for the experiments.

CP-VNT_CREATE Workflow

Figure 5 shows the relevant messages for the CP-VNT_CREATE workflow that is triggered by the operator through the NMS. Its goal is to add the set of ICs belonging to the CP-VNT and the LCs authorized to connect to the CP-VNT to the ASO's database and to set-up the CP-VNT itself. For the latter, the ASO issues a CP-VNT creation request message (labelled as 1 in Fig. 5) to the CTRL with the set of ICs to be connected. The CTRL issues a PCRequest (PCReq) message to the fPCE (2), which delegates the CP-VNT topology computation to the bPCE (3). The bPCE runs the CP-VNT_CREATE algorithm and replies the solution in a PCReply (PCRep) message (4). The fPCE issues a single PCInitiate (PCInit) message to the PM containing all the LSPs to be implemented and waits for the PCReport (PCRpt) message confirming the implementation of each of the lightpaths (5), i.e. one PCRpt per lightpath. When the CP-VNT topology has been created, the fPCE sends back a PCReply (PCRep) message to the CTRL (6) with the lightpaths implemented, which replies to the ASO (7) that updates its services database and informs the NMS.

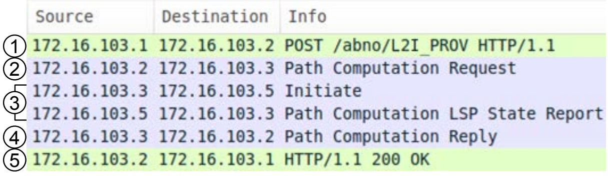
L2I-PROV Workflow

Figure 6 shows the relevant messages for the L2I-PROV workflow. When a LC needs to synchronize contents with the ICs, the local cloud resource manager issues a L2I connection request to the ASO. The ASO checks whether the LC belongs to the CP and issues a message to the CTRL with the request (message 1 in Fig. 6). The CTRL creates a PCReq message and forwards it to the fPCE (2), which computes the route for the connection using the L2I-PROV algorithm and issues a PCInit message for the LSP to the PM. When the LSP has been set-up, the fPCE replies to the CTRL (4) that in turn replies to the ASO (5). Finally, the ASO updates its database and confirms to the cloud resource manager the availability of the L2I connection.



	Source	Destination	Info
①	172.16.103.1	172.16.103.2	POST /abno/CPVNT_CREATE HTTP/1.1
②	172.16.103.2	172.16.103.3	Path Computation Request
③	172.16.103.3	172.16.103.4	Path Computation Request
④	172.16.103.4	172.16.103.3	Path Computation Reply
	172.16.103.3	172.16.103.5	Initiate
⑤	172.16.103.5	172.16.103.3	Path Computation LSP State Report
	172.16.103.5	172.16.103.3	Path Computation LSP State Report
⑥	172.16.103.3	172.16.103.2	Path Computation Reply
⑦	172.16.103.2	172.16.103.1	HTTP/1.1 200 OK

Figure 5. CP-VNT_CREATE message list.



	Source	Destination	Info
①	172.16.103.1	172.16.103.2	POST /abno/L2I_PROV HTTP/1.1
②	172.16.103.2	172.16.103.3	Path Computation Request
③	172.16.103.3	172.16.103.5	Initiate
	172.16.103.5	172.16.103.3	Path Computation LSP State Report
④	172.16.103.3	172.16.103.2	Path Computation Reply
⑤	172.16.103.2	172.16.103.1	HTTP/1.1 200 OK

Figure 6. L2I-PROV message list.

5. CONCLUDING REMARKS

A hierarchical architecture for video distribution was studied. IC nodes were geographically distributed and interconnected to create a specific VNT for each CP. Contents in the IC nodes are automatically synchronized using the connectivity provided by the CP-VNT, whereas LC nodes synchronize their contents with IC nodes periodically. To that end, L2I anycast connections are dynamically requested. Two main problems were identified for operating this architecture: the CP-VNT creation and the L2I connection provisioning. These problems were formally stated and heuristic algorithms were eventually proposed for solving them. Exhaustive simulations were carried out to evaluate the performance of the architecture. The results showed that high efficiency can be achieved when increasing the number of IC nodes in the VNT; more than 66% load increment was observed. Aiming at experimentally assessing the proposed architecture an ABNO-based control plane architecture was considered. The ASO module was responsible for service management and issues requests to the ABNO. Workflows were developed for the identified problems and an experimental assessment was carried out in our experimental SYNERGY test-bed running iONE software.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the Spanish MINECO SYNERGY project (TEC2014-59995-R) and from the Catalan Institution for Research and Advanced Studies (ICREA).

REFERENCES

[1] "Metro Network Traffic Growth: An Architecture Impact Study," Alcatel-Lucent Bell Labs strategic white paper, Dec. 2013 [Online]. Available: http://resources.alcatel-lucent.com/asset/171568.

[2] L. M. Contreras *et al*., "Toward cloud-ready transport networks," *IEEE Communications Magazine*, vol. 50, pp. 48-55, 2012.

[3] L. Velasco *et al.,* "A service-oriented hybrid access network and cloud architecture," *IEEE Communications Magazine*, vol. 53, pp. 159-165, 2015.

[4] A. Asensio *et al*., "Scalability of telecom cloud architectures for live-TV distribution," in *Proc. OSA Optical Fiber Communications Conference (OFC)*, 2015.

[5] L. Velasco *et al*., "Solving routing and spectrum allocation related optimization problems: From off-line to in-operation flexgrid network planning [Invited Tutorial]," *IEEE/OSA Journal of Lightwave Technology*, vol. 32, pp. 2780-2795, 2014.

[6] L. Velasco *et al*., "Cross-stratum orchestration and flexgrid optical networks for datacenter federations," *IEEE Network Magazine*, vol. 27, pp. 23-30, 2013.

[7] L. Velasco and Ll. Gifre, "iONE: A workflow-oriented ABNO implementation," in *Proc. IEEE/OSA Photonics in Switching Conference (PS)*, 2015.