Assessing the Performance of Multi-Layer Path Computation Algorithms for different PCE Architectures

S. Martínez[1*], V. López[1], M. Chamania[2], O. González[1], A. Jukan[2], J.P. Fernández-Palacios[1]
[1] Telefónica I+D, Madrid, Spain [2] Technical Technische Universität Carolo-Wilhelmina zu Braunschweig
**e-mail: smta@tid.es*

Abstract: We have implemented a multi-layer PCE and compared performance of various algorithms using either one integrated, or two separate PCEs in each layer. Multi-layer integration reduces blocking by 13,91%, but increases computation time by 49,24%.

OCIS codes: (060.0060) Fiber optics and optical communications; (060.4256) Networks, network optimization.

1. Introduction

To address the increasing service demand in an efficient fashion, operators are considering the multi-layer architectures, with IP routing and optical switching working together [1]. A typical core network consists of an IP/MPLS network designed over a Wavelength-Switched Optical Network (WSON) network. In a combined network infrastructure, the IP/MPLS links are made up of Lambda switching Capable (LSC) Label Switched Paths (LSPs) established in the transport network based on optical transmission, while connections/services are established using Packet Switching Capable (PSC) LSPs in the IP/MPLS network. Here, a joint path computation is essential, which can be implemented within the already existing Path Computation Element (PCE) framework.

There are basically two different schemes proposed to compute multi-layer paths with the PCE [2]: (i) integrated (single) multi-layer PCE where a unique PCE has the complete multi-layer topology information, a centralized PCE server uses topology information available in both Traffic Engineering Databases (TEDBs) to compute optimal paths. (ii) coordinated (multiple) PCEs solution where each network layer has its own PCE that can communicate with each other when required to compute multi-layer paths. The exchange of requests allows the IP/MPLS network layer to request a path segment in the optical (server) layer to optimize the provisioning of one or more connections requested in the client layer. In our previous work [3], we presented an implementation of the multiple cooperating PCEs, and in this work we implement the single multi-layer PCE and various path computation algorithms, such as proposed in [4], for both integrated and coordinated PCE solutions to compare the performance in terms of blocking and computation delay for both PCE architectures.

2. The Multi-Layer Algorithms Implemented to Test

The primary challenge of multi-layer path computation algorithms is to set up new LSPs in different layers while taking into account different layer-specific constraints. Multi-layer algorithms in a typical network topology should be able to (1) route lightpaths over the physical topology (WSON layer), by calculating new LSPs, and assign wavelengths to generate the client network topology in the IP layer, and (2) route connections in the client network topology. According to [5], multi-layer path computation for a request in the client network can be reduced to a combination of one or more of 4 different operations: (1) Operation 1 attempts to route a connection in the client network over an existing lightpath directly between the source and the destination. This operation is equivalent to computing a 1-hop path in the client network for a request. (2) Operation 2 attempts to allocate the request in the client network only, i.e. by only using existing lightpaths. (3) Operation 3 creates a new lightpath from source to destination to route the request, and in this operation an LSP is computed in the transport network topology from the

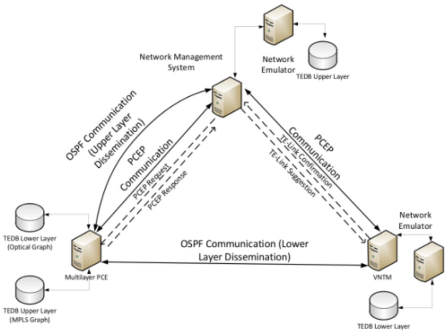


Fig. 1. Telefónica emulator architecture with ML-PCE

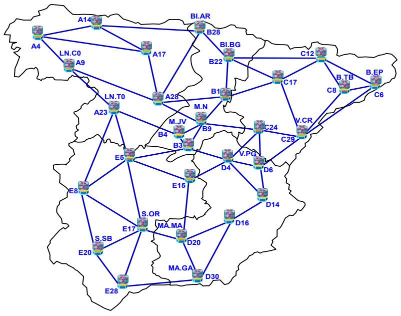


Fig. 2. Telefónica Spanish Backbone Network

source to the destination. (4) Finally, operation 4 routes the traffic over existing lightpaths and creates one or more lightpaths between intermediate nodes to route a connection from a source to a destination. Based on these operations, authors in [5] identify three different policies for multi-layer path computation:

- **Minimizing the number of Traffic Hops (MinTH).** This policy firstly uses Operation 1 to compute an existing single-hop path and if it fails, a new end-to-end lightpath is created (Operation 3). If both fail, paths are computed using Operations 2 and 4 and the one with fewer hops on the virtual topology is chosen.
- **Minimizing the number of Light-Paths (MinLP).** The policy first uses Operation 1, and if unsuccessful Operation 2 is run to compute a path in the virtual topology. In case of failure, Operation 3 and later Operation 4 are performed to compute a path.

3. Testbed set-up

In our testbed set-up, we compare the performance of the integrated multi-layer PCE and the coordinated multi-layer PCE framework with a real world PCE implementation (via emulations). Fig. 1 shows the reference architecture for the integrated multi-layer PCE scenario. The physical network used is the Spanish Backbone Network (Fig. 2), with each link supporting 80 wavelengths (realistic assumption). We implement basic network emulators to maintain reservation status, statistics and update the TEDs. A Network Management System (NMS) generates 5000 connection requests with random source-destination selection and random bandwidth selection between 1 and 10Gbps. For the purpose of this study, all connections, once established, are assumed to be permanent to check the PCE performance and grooming behavior in a non-dynamic scenario. The request from the NMS is sent to the ML-PCE which can compute single or multi-layer paths. The multi-layer path is sent in the form of a multi-layer ERO, an example of which is Fig. 6. The multi-layer ERO is constructed using standard ERO objects to represent paths segments in individual layers, and use the SERVER_LAYER_INFO sub-object to indicate a layer change. For the example shown in Fig. 6, the path is computed as an end-to-end lightpath (Operation 3) from a source (192.168.8.3) to destination (192.168.8.1) in the MPLS network. After the source node, a SERVER LAYER INFO sub-object is included to highlight the layer change. The next sub-objects are unnumbered interfaces as well as Label control, which identify the lambda in the optical domain. Once the optical path is defined in the ERO, another SERVER LAYER INFO sub-object is included before the destination to identify the change to the upper layer. In case of a multi-layer path, the NMS must first establish the lightpaths before establishing the connections, and requests the Virtual Network Topology Manager (VNTM) for the same by forwarding the ERO inside a TE-Link Suggestion message to the VNTM [5]. If the lightpaths are established successfully, the NMS and the VNTM send OSPF LSA messages to the PCE to update the PCE TED. For the multiple PCE scenario, all other elements remain unchanged but instead of a single PCE, there are 2 PCEs. The NMS sends a request to the MPLS PCE, which if required communicates with the WSON PCE to compute a path. The computed WSON LSPs in are then included by the MPLS PCE in a multi-layer ERO and sent to the NMS.

To run the aforementioned policies in a single TED, authors in [4] propose to construct an auxiliary graph with N+1 layers, where N is the number of wavelengths (80 for this experiments). The edge weights in this auxiliary graph are defined for each policy [4] and the K-Shortest Path (KSP) algorithm is used to compute a path. For the two TEDBs case, Operations 1,2 and 3 can be computed by individual PCEs (MPLS or WSON) while for Operation 4, the MPLS PCE generates suggestions for ingress and egress MPLS routers between which an LSP should be established, and the WSON PCE then attempts to compute a lightpath between the suggested ingress-egress nodes.

4. Algorithms evaluation

Fig. 3 shows the number of blocked connections for MinLP and MinTH policies in the case of a single or two TEDBs. For any policy, we see that a single TEDB reduces the number of blocked connections in the network. For example, in the case of the MinTH policy, the first blocked connection is observed at request 889 when multiple PCEs are used and at request 1362 for a single PCE case, while the corresponding numbers for the MinLP policy are 962 and 1101 connections respectively. Fig. 4 shows the allocated bandwidth for both policies in the case of a single or multiple PCEs, and we see that the MinTH policy can improve total capacity allocation by 13.91% when using a single PCE as compared to multiple PCEs while the increase in the MinLP policy is 9,97%.

The integrated PCE contains the complete topology information and as a result can optimize resource allocation for any algorithm, when compared with the multiple PCE. The coordinated PCE on the other hand increases the utilization of the optical resources in the network: the MinTH policy uses 64,3% and 60,2% of the wavelengths for single and multiple PCE scenario respectively, while the same values for the MinLP are 63,1% and 58,9%, due to the preference to minimize new lightpaths.

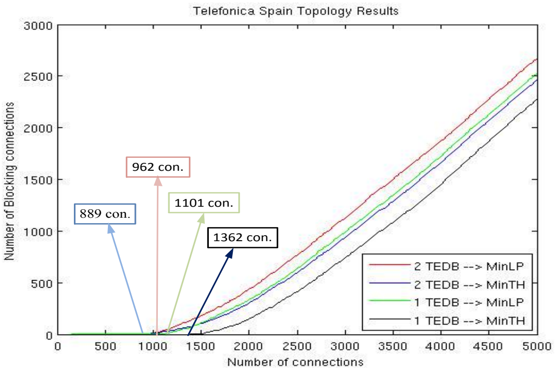


Fig. 3. Number of blocking connections

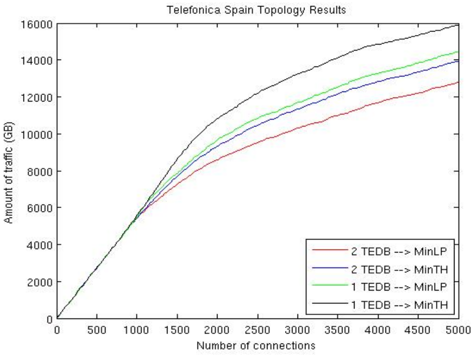


Fig. 4. Amount of allocated bandwidth

We also compare the mean computation time taken for both policies in case of single and Multiple PCEs in Fig. 5. The computation on the Single PCE has to work with a large topology and as a result needs 54.32ms and 53.57ms for MinLP and MinTH policies, respectively, while the multiple PCE solution computes connections for the same policies in 29.15ms and 27.19ms respectively, as the computation of the paths in individual layers have significantly lowered complexity than the integrated multi-layer path computation. Note that both PCEs are implemented on the same host for this specific scenario.

In light of the results, we can conclude that there is an important increment in computation time for integrated multi-layer PCE solution. Here, the PCE is running KSP in a large graph, which has N+1 layers (81 in this case). As KSP complexity increases linearly with number of arcs and nodes the computation time for this algorithm is much higher than in the case of multiple PCEs, where each of PCEs runs their own KSP is run. Note that in the case of multiple coordinated PCEs, First Fit is run for wavelength assignment once KSP finds candidate paths.

5. Conclusions

Multi-layer coordination is becoming an important requirement in network operators, due to its potential to reduce carrier's investments, network interventions and thus cost of operations. Multi-layer PCE is a key technology to study for this scenario. This work shows the performance achieved with full topological information or with partial information. To do so, we have implemented a state-of-the-art protocol in Telefonica PCE test-bed and compare the performance. Results show that integrated TED PCE can increase 13,91% bandwidth utilization in Spanish topology, but its computational time increases 49,24% in comparison with two coordinated TEDs per layer.

Acknowledgements

This work has received funding from the EC FP7 Programme ICT STRONGEST under grant agreement no 247674 and the ONE project in contract number INFSO-ICT-258300.

References

[1] J. E. Gabeiras, et al., "Is multilayer networking feasible?", Optical Switching and Networking, vol. 6, no. 2, pp. 129-140 (April, 2009).
[2] V. López, et al., "Path computation element in telecom networks: Recent developments and standardization activities," in Proc. ONDM, 2010.
[3] M. Chamania, et al., "Coordinated Computation of Multi-layer Paths via Inter-layer PCE Communication: Standards, Interoperability and Deployment," in IEEE ICC workshop on Telecommunications: From Research to Standards, Jun 2012.
[4] H. Zhu, et al., "A novel generic graph model for traffic grooming in heterogeneous WDM mesh networks," IEEE/ACM Trans. , 2003.
[5] O. Gonzalez de Dios, et al., "Functional validation of the cooperation VNTM and PCE", in Proc. iPOP, Jun 2012

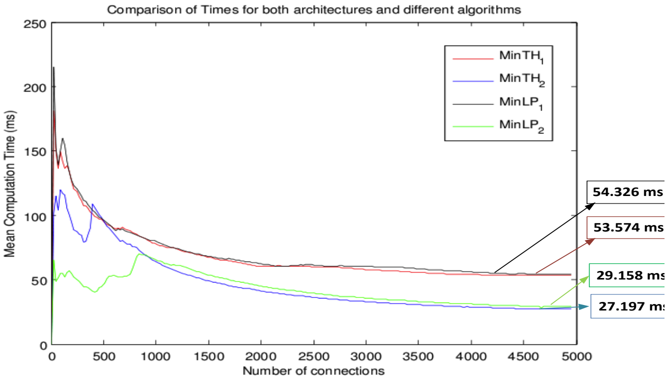
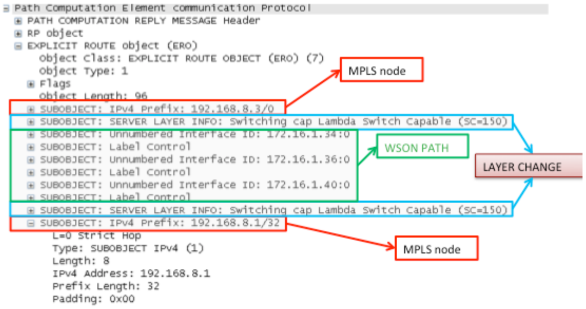


Fig. 5. Mean computation time (ms)



```
⊟ Path Computation Element communication Protocol
  ⊞ PATH COMPUTATION REPLY MESSAGE Header
  ⊞ RP object
  ⊟ EXPLICIT ROUTE object (ERO)
      Object Class: EXPLICIT ROUTE OBJECT (ERO) (7)
      Object Type: 1
    ⊞ Flags
      Object Length: 96
    ⊞ SUBOBJECT: IPv4 Prefix: 192.168.8.3/0
    ⊞ SUBOBJECT: SERVER LAYER INFO: Switching cap Lambda Switch Capable (SC=150)
    ⊞ SUBOBJECT: Unnumbered Interface ID: 172.16.1.34:0
    ⊞ SUBOBJECT: Label Control
    ⊞ SUBOBJECT: Unnumbered Interface ID: 172.16.1.36:0
    ⊞ SUBOBJECT: Label Control
    ⊞ SUBOBJECT: Unnumbered Interface ID: 172.16.1.40:0
    ⊞ SUBOBJECT: Label Control
    ⊞ SUBOBJECT: SERVER LAYER INFO: Switching cap Lambda Switch Capable (SC=150)
    ⊟ SUBOBJECT: IPv4 Prefix: 192.168.8.1/32
        L=0 Strict Hop
        Type: SUBOBJECT IPv4 (1)
        Length: 8
        IPv4 Address: 192.168.8.1
        Prefix Length: 32
        Padding: 0x00
```

Fig. 6. Wireshark trace of PCEP Reply message