Experimental Demonstration of an Enhanced Impairment-Aware Path Computation Element

S. Spadaro[1], J. Perelló[1], F. Agraz[1], S. Azodolmolky[1,2], M. Angelou[1,2], Y. Qin[3], R. Nejabati[3], D. Simeonidou[3], P. Kokkinos[4], E. Varvarigos[4], Y. Ye[5] and I. Tomkos[2]

(1)Universitat Politècnica de Catalunya (UPC), Barcelona, Spain; (2) Athens Information Technology (AIT), Athens, Greece,
(3) University of Essex, Colchester, United Kingdom; (4) Research Academic Computer Technology Institute (RACTI), Patras, Greece;
(5) Huawei Technologies, Munich, Germany; e-mail address: spadaro@tsc.upc.edu

Abstract: An enhanced impairment-aware path computation element (EPCE) for dynamic transparent optical networks is proposed and experimentally evaluated. The obtained results show that by using the EPCE, light-path setup times of few seconds are achieved.

OCIS codes: (060.4250) Networks; (060.4510) Optical Communications

1. Introduction

Considering the impact of physical layer impairments in the planning and operation of all-optical networks has been the main goal of the EU DICONET Project [1]. Essentially, the main outcome of this project has been the development of a dynamic Network Planning and Operation Tool (NPOT) that incorporates real-time assessments of the optical layer performance into Impairment Aware Routing and Wavelength Assignment (IA-RWA) algorithms.

This NPOT module is integrated into a Generalized Multi-Protocol Label Switching (GMPLS)-based control plane. In fact, two NPOT/GMPLS integration schemes have been defined within the DICONET project, namely, distributed and centralized. As concluded in [2], the centralized integration scheme yields lower network Blocking Probability (BP), but with the cost of increased light-path setup time. In this work, we propose the integration of the centralized NPOT presented in [2] with a standard Path Computation Element (PCE). The resulting centralized entity is called Enhanced PCE (hereafter EPCE). The EPCE implements the standard Path Computation Element Protocol (PCEP) to allow route requests from the network Optical Connection Controllers (OCCs), in addition to an extended OSPF-TE protocol for the dissemination of wavelength availability and Physical Layer Information (PLI).

In this paper we experimentally evaluate the performance of the proposed EPCE in a dynamic all-optical network environment. From the experimental results, the reported EPCE shows similar setup delays as the distributed scheme, while still keeping the low BP figures of a centralized one.

2. Enhanced Path Computation Element

Fig. 1 shows the architectural block diagram of the centralized control plane scheme designed in the framework of the EU DICONET project. In this architecture, the EPCE is responsible for the IA-RWA computation, while the OCCs run the extended GMPLS OSPF-TE protocol, a standard RSVP-TE implementation and the interface to the actual optical nodes. The role of the standard PCE within the proposed EPCE is twofold. Firstly, it participates in the standard PCEP messages exchanges with OCCs. Secondly, it implements a TCP-based communication with the NPOT for requesting IA-RWA path computations. The NPOT maintains and manages the global topology and physical parameters repositories. These repositories are named global Traffic Engineering Database and global Physical Parameters Database (gTED and gPPD), respectively. The light-path computation inside the NPOT is performed utilizing the multi-parametric IA-RWA algorithm presented in [3].

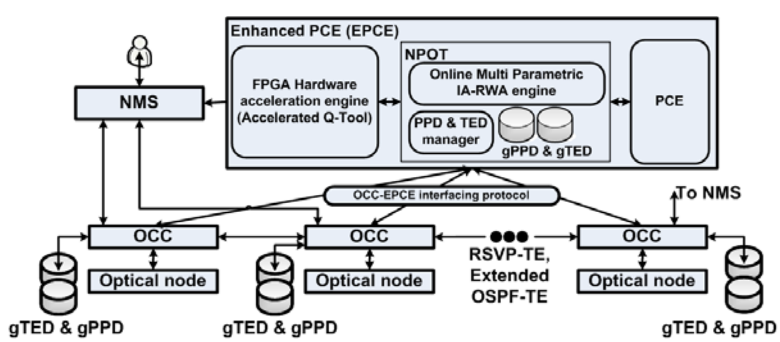

Fig. 1. The Enhanced PCE (EPCE).

Upon the arrival of a new connection request, the source OCC contacts the EPCE to request an impairment aware light-path computation. During the light-path computation, the online IA-RWA module of NPOT utilizes the QoT estimator (Q-Tool) module and the information of the gPPD and gTED. When the EPCE finds a light-path with guaranteed QoT (Q-factor value above a predefined threshold), the light-path is returned back to the source OCC which triggers the standard RSVP-TE signaling protocol. Upon successful establishment of a light-path, the global PPD and TED in the EPCE are updated using the extended OSPF-TE protocol. Finally, the source OCC updates the Network Management System (NMS). In case of lack of resources, or unacceptable QoT, the demand is blocked and the source OCC also informs the NMS accordingly.

The computationally intensive algorithms inside the Q-Tool can make the QoT estimator be the bottleneck in the impairment-aware route computation. In view of this, a Field Programmable Gate Array (FPGA)-accelerated QoT estimator is also integrated inside the EPCE. The FPGA-accelerated Q-Tool is implemented in a Xilinx Virtex IV FPGA with an embedded processor. The Q-Tool algorithms have been implemented using FPGA hardware logics to achieve the lowest Q-factor computation time. The communication protocol with the NPOT has been implemented utilizing the embedded processor of the FPGA (an IBM Power PC running at 300 MHz). The performance of the FPGA-based QoT estimator module has been reported in [4].

3. Experimental Evaluation

The performance of the proposed EPCE has been evaluated on the DICONET test-bed located at the UPC premises in Barcelona. This testbed describes a realistic 14-node transport network scenario (Fig. 2) with 10 bidirectional wavelengths at 10 Gbps per link. Each network node is composed of an OCC and a WSS-based OXC emulator, both interconnected through the Connection Controller Interface (CCI). All OCCs are also connected to the EPCE through the OCC-EPCE interface. The connectivity between OCCs is supported over 100 Mbps point-to-point Ethernet links, which describe an out-of-fiber control plane with the same topology as the emulated all-optical data plane. OCCs implement the standard RSVP-TE protocol and an extended version of the OSPF-TE protocol to disseminate the PLI and wavelength availability information. Uniformly distributed light-path requests arrive to the network following a Poisson process. The light-path holding times (HTs) are exponentially distributed with mean 600 seconds. Different loads are generated by adjusting the light-path inter-arrival times (IATs) accordingly (load = HT/IAT). Fig. 3 depicts the average light-path setup time in the network, depending on whether the Q-Tool in the NPOT is software-based (non-accelerated) or FPGA-accelerated. These results are also benchmarked to the setup times obtained by the impairment-aware distributed light-path provisioning approach reported in [2]. As seen, the light-path setup time strongly increases with the offered load when the software based Q-Tool is used. Note that no route computation concurrency is allowed in the NPOT. Moreover, a sufficient amount of time must be left between two consecutive route computations in order to let the centralized NPOT be fed with the new physical layer and wavelength availability information. Hence, the NPOT scheduler must delay incoming requests until the computation, signaling and OSPF-TE flooding of the previous connection establishment are completed. Although waiting for the OSPF-TE flooding completion between route computations might be identified as conservative, it really serves the purpose of minimizing the network BP. Conversely, the distributed approach allows parallel light-path establishments, as the Q-factor values of the light-path to be established and the potentially affected ones are computed during the signaling process [2].

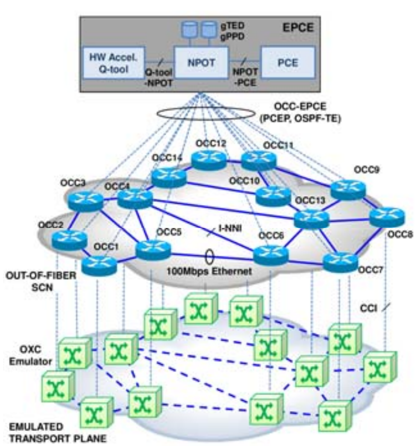

Fig. 2. The DICONET experimental test-bed.

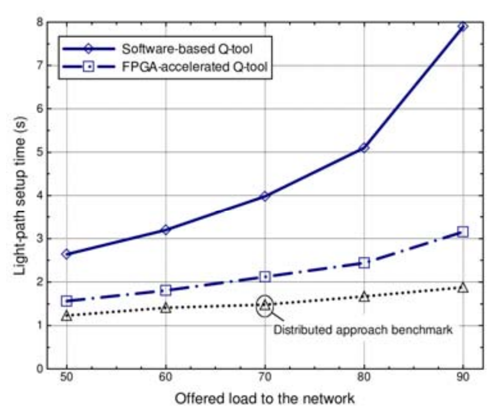

Fig. 3. Average light-path setup time vs. offered load (Erlang).

Interestingly, the FPGA-accelerated Q-Tool allows the EPCE to show similar setup delays as the distributed approach, while still keeping the low BP figures of a centralized impairment-aware route computation with fully updated physical layer and wavelength availability information [2]. For instance, for a medium load of 70 Erlang, the FPGA acceleration allows to reduce the setup time from 4 seconds to 2.21 seconds (45% reduction), quite close to the 1.5 s of the distributed approach. Furthermore, the BP with the EPCE is 3% compared to the 4.4% in the distributed case. For higher loads, even more pronounced setup time reduction is observed, i.e., 60% improvement for 90 Erlang. Fig. 4 breaks down the different contributions to the light-path setup time. Specifically, the waiting time in the NPOT scheduler, the NPOT processing time, the Q-factor computation time and the control plane signaling time have been considered. The NPOT processing time comprises the total amount of time needed to process the route request, trigger the IA-RWA algorithm and send the Explicit Route Object (ERO) back to the source node OCC. Leaving the waiting and signaling times aside (the latter, around 30 ms, is almost imperceptible in the graph), the Q-factor computation arises as the route computation bottleneck in case of the software-based Q-Tool. In fact, it becomes 1.43 s for 90 Erlang, compared to the 570 ms of the NPOT processing time. In contrast, the FPGA-accelerated Q-Tool enables much faster Q-factor estimation, scaling also perfectly with the offered load to the network. As shown, the Q-factor computation time only raises from 190 ms to 218 ms when increasing the offered load from 50 to 90 Erlang, that is, only a 15% increment while almost doubling the offered load to the network. In particular, in a medium-loaded scenario with 70 Erlang, the average Q-factor computation time can be decreased from 1.16 s to 210 ms with the FPGA acceleration (x5.5 faster). On average, the Q-Tool is invoked to compute the Q-factor value of about 10 light-paths simultaneously. Fig. 5 plots the Cumulative Distribution Function (CDF) and relative frequency of the FPGA-accelerated Q-factor computation time for the same offered load. Evidently, around 80% of the Q-factor computation invocations experience a response time below 300 ms, while no invocation goes beyond 500 ms. Note that the measured light-path setup times remain drastically below the 10 s of the impairment-aware network management system presented in [5]. As a matter of fact, having the NPOT scheduler empty, setup times < 1 s would be obtained in every evaluated load scenario.

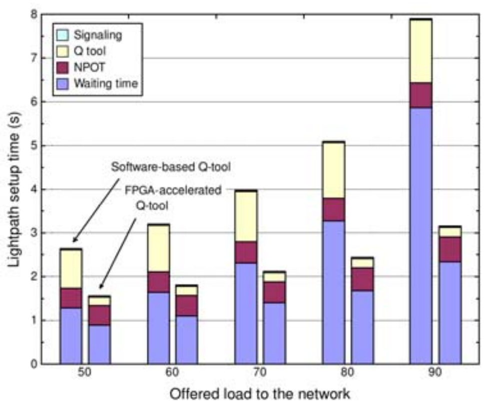

Fig. 4. Light-path setup time breakdown without (left bars) and with (right bars) FPGA acceleration.

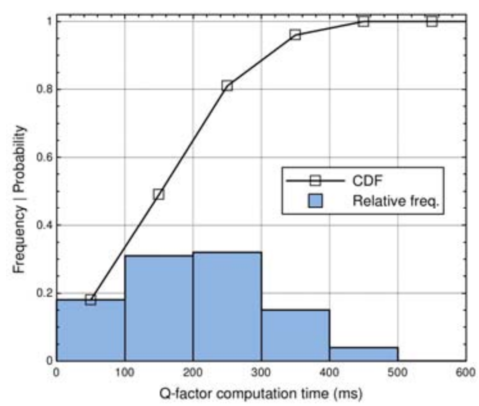

Fig. 5. Relative frequency and CDF of the Q-factor computation time with FPGA-acceleration (load=70 Erlang).

4. Conclusions

This work demonstrated an enhanced PCE for IA-RWA for transparent optical networks. The overall light-path setup times obtained through the integration of the EPCE and an extended GMPLS-based control plane is much lower than the times previously reported. Specifically, if the EPCE is provided with a hardware-based accelerated QoT estimator, light-path setup times below 3 seconds are obtained, even in highly loaded network scenarios. This work has been partially supported by the Spanish Science Ministry through the project ENGINE (TEC2008-02634).

4. References

[1] S. Azodolmolky, et al., “A dynamic impairment aware networking solution for transparent meshed optical networks”, IEEE Communications Magazine, vol. 47, no. 5, pp. 38-47, 2009.

[2] F. Agraz et al., “Experimental Demonstration of Centralized and Distributed Impairment-Aware Control Plane Schemes for Dynamic Transparent Optical Networks”, in Proc. of OSA OFC/NFOEC 2010, PDPD5, 2010.

[3] P. Kokkinos et al., “Multi-Parametric Online RWA Based on Impairment Generating Sources”, in Proc. of IEEE Globecom 2009, Honolulu, Hawaii, USA, 2009.

[4] Y. Qin et al., “Demonstration of C/S based Hardware Accelerated QoT Estimation Tool in Dynamic Impairment-Aware Optical Network”, in Proc. of 36th European Conference and Exhibition on Optical Communication (ECOC 2010), P5.17, Torino, Italy, September 2010.

[5] T. Tsuritani et al., “Optical path computation element interworking with network management system for transparent mesh networks”, in Proc. of OSA OFC/NFOEC 2008, NWF5, 2008.