Hardware Accelerated Impairment Aware Control Plane

Yixuan Qin[1], Yvan Pointurier[2], Eduard Escalona[1], Siamak Azodolmolky[2], Marianna Angelou[2], Ioannis Tomkos[2], Kostas Ramantas[3], Kyriakos Vlachos[3], Reza Nejabati[1] and Dimitra Simeonidou[1]

[1] *CES Department, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, UK*
[2] *AIT P.O. Box 68, 0.8km Markopoulo Ave., 19002 Peania, Greece*
[3] *CEI Department, University of Patras, 26500 Rio, Patras, Greece*
{yqin, eescal, rnejab, dsimeo}@essex.ac.uk, {yvan, sazo, mang, itom}@ait.edu.gr, {ramantas, kvlachos}@ceid.upatras.gr

Abstract: This paper presents a hardware accelerated QoT estimation tool used in the DICONET impairment-aware optical network. Performance evaluation is given by examining different network scenarios in terms of network size and number of wavelengths.

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

1. Introduction

Emerging network applications and services have stringent requirements in terms of bandwidth and quality of service (QoS) which can be met by optical networks. However, in optical networks, physical layer impairments inject extra constraints on QoS of transmission services. To address this need in a scalable manner, a dynamic impairment aware (IA) optical network is desired. This dynamicity is provided by an IA control plane (CP) that enables real time candidate lightpath computation to support optimum network traffic engineering under dynamically changing traffic and physical network conditions. To approach this goal, a proprietary dynamic network planning and operation tool (NPOT) has been developed in the DICONET project [1]. It integrates advanced physical layer models with novel routing and wavelength assignment (RWA) algorithms to provide a mechanism for physical layer impairment aware transport network provisioning. Upon a connection request, the optical signal quality of transmission (QoT) is estimated in order to aid the NPOT to make the decision of accepting or rejecting candidate lightpaths' establishment. However, the complex algorithmic calculations performed for QoT estimation dramatically increase the overall lightpath setup time, making it unsuitable for real time dynamic lighpath provisioning.

This paper, presents a hardware-accelerated QoT estimation tool which sharply decreases the lightpath assessment time and accelerates lightpath provisioning with the objective of making it suitable for incorporation in a dynamic and real-time environment such as Generalized MultiProtocol Label Switched (GMPLS) controlled optical networks. The hardware-accelerated tool enables timing critical applications which need fast response and low delay. Hence, it also facilitates finer granularity of resource sharing and better resource utilization when the signalling protocol is able to set-up and tear-down connections more often. Furthermore, a faster route calculation reduces dramatically the possibility of finding resource availability inaccuracy due to unadvertised changes during the QoT estimation.

2. IA CP architecture and performance bottleneck

In DICONET architecture, an extended GMPLS CP is deployed, based on a path computation element (PCE) centralised approach. The PCE receives the path calculation request (*e.g.* from the Network Management System); it is then forwarded to the NPOT which is responsible for the actual path computation and the QoT estimation in particular. IA CP function block interaction is shown in Fig. 1. In this model, the OSPF-TE IGP routing protocol is extended to disseminate the physical layer impairment information through the network. The PCE communication protocol (PCEP) is the protocol used to communicate between the PCE and the path computation client (PCC) and between PCEs for inter-domain path calculation. Upon the result retrieved from QoT estimation, if there is an available path, the information is transferred back to the optical connection controller (OCC) via PCEP, and then standard RSVP-TE messages are triggered to set-up the lightpath. If the path reservation fails or the QoT estimation tool gives negative Q factor for alternative paths, the Network Management System is notified accordingly.

The current implementation of the QoT estimation tool is purely software based, which is reasonable for the preliminary module development stage. Based on the RWA performance tests reported in [2], the running time is dominated by the execution time of the QoT estimation and can be in the range from 10 to 1000 seconds depending on

978-1-55752-884-1/10/$26.00 ©2010 IEEE

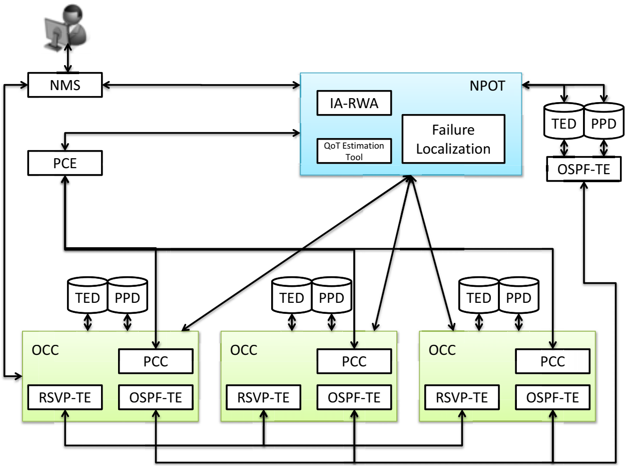


Fig. 1: Centralized/PCE based CP

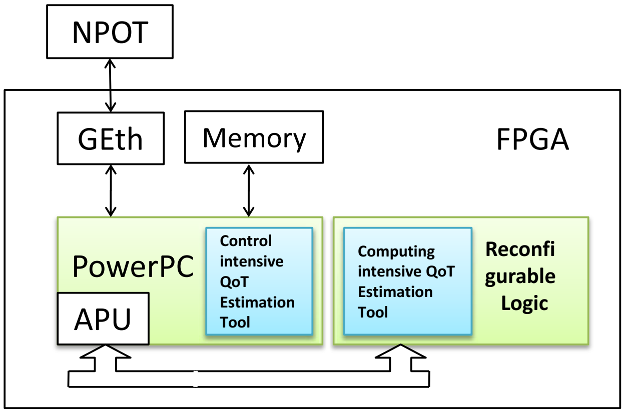

NPOT
GEth
Memory
FPGA
PowerPC
Control intensive QoT Estimation Tool
APU
Computing intensive QoT Estimation Tool
Reconfigurable Logic

Fig. 2: Hardware Accelerated QoT estimation tool

the network load and RWA algorithms deployed. This performance is not suitable for practical lightpath connection request which is in the order of seconds not to mention hardware accelerated signalling mechanisms which can achieve the magnitude of microseconds [3]. In the DICONET approach, path computation in GMPLS depends directly on QoT estimation results; a slow estimation reduces the performance of GMPLS significantly.

3. Hardware accelerated QoT estimation tool implementation

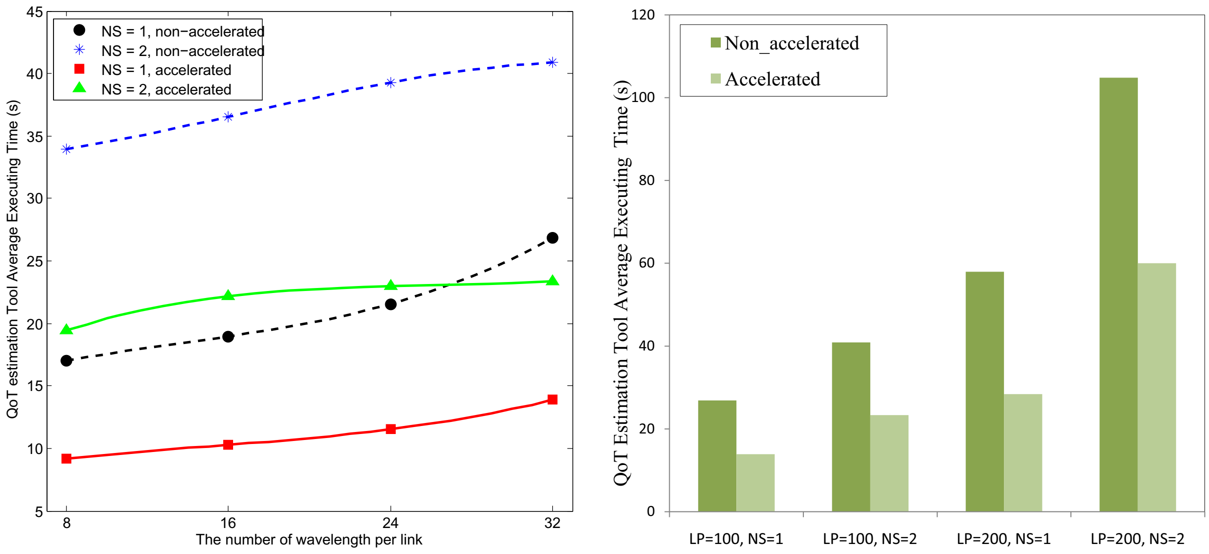
As for the challenge posed by the complexity of the QoT estimation tool, our approach focuses on the implementation of only the time-critical operations of the Q factor computation in hardware (a Xilinx Virtex 4 FPGA), and keep the nontime-critical operations in software (running at an IBM PowerPC 405 hard core 300MHz which is embedded inside the FPGA fabric, in conjunction with 1GB DDR2 memory). Another reason to separate the QoT estimation tool is that the FPGA achieves high performance results when executes computing intensive operations, but is overshadowed by control intensive operations while the converse is true for general purpose CPUs. The data exchange between the reconfigurable logic in the FPGA fabric and the pipeline of integrated PowerPC is achieved by deploying the auxiliary processor unit (APU), a flexible high-bandwidth interface as shown in Fig. 2. One of the challenges to accelerate QoT estimation tool is to accurately identify computationally-intensive operations and offload them to the FPGA via the APU interface, in order to significantly improve the hardware-based QoT estimation tool performance.

In this work, GNU software "gprof" [4] is used to help find the most CPU time-consuming routines. Then, this information is used as a reference to carry out a careful design. In order to achieve the highest performance, a two-level acceleration scheme is proposed — at instruction level, and at routine level. The former is mainly used for mathematic computation, *i.e.* arrays and complex number operations. The latter is used when many individual instruction level accelerations need to be done; then the APU interface overhead becomes significant. In this case, the whole routine can be migrated into the FPGA reconfigurable logic if it is allowed by the FPGA resource, and the APU will only be invoked once when returning the results.

4. Results and performance study

The non-accelerated QoT estimation tool is evaluated in a very fast general purpose computer which populated with an Intel Quad Core Extreme 3.2GHz CPU and 4GB DDR3 memory, while the accelerated version runs in the FPGA evaluation board mentioned in previous section. The experiments carried out in this work are based on a national network topology provided by Deutsche Telekom (DT). Two variations ("network sizes") of the DT topology have been used: one with original link lengths, and the other where all link lengths are doubled. In addition the QoT estimation tool performance is evaluated in terms of the number of available wavelength per link, varying from 8 to 32 with a step of 8, and for a varying number of lightpaths established in the network. The QoT estimator includes single channel effects (ASE noise, PMD, SPM and chromatic dispersion) and multi-channel effects (XPM and FWM), such that all of the network size, number of channels and number of established lightpaths impact on its running time.

Given the above scenarios, Fig. 3a shows the QoT estimation tool performance for 100 lightpaths; we report the average executing time against the number of wavelength per link for the two different network sizes (NS={1,2}). For



(a) (b)

Fig. 3: The QoT estimation tool average executing time in two different network sizes *v.s.* (a) the number of wavelength per link for 100 lightpaths and (b) the number of lightpaths for 32 wavelengths per link

the original network size (NS=1), the non-accelerated QoT estimation tool execution time (dotted line with circle) is ranging from 17s to 27s; the corresponding accelerated computation (solid line with square) is nearly halved, a very significant speed-up. When network size is doubled, generally the computation time is doubled as well. The non-accelerated QoT estimation tool takes 34s to 41s to run depending on the number of available wavelengths per link; while the accelerated QoT estimation tool execution time drops down to about half of this. In the worst case (23s) the accelerated tool is still much faster than best-case non-accelerated tool.

The QoT estimation tool performance for combinations of two numbers of lightpath (LP={100,200}) and two network sizes (NS={1,2}) is shown in Fig. 3b for a fixed number of wavelengths per link (32). As shown in the figure, the accelerated QoT estimation tool outperforms the non-accelerated one by almost 100% for each combination of LP and N; it scales smoothly and mitigates the impact of increasing load. It is a very important desirable feature for a QoT estimation tool in a real dynamic network to provide a consistent network provisioning delay.

5. Conclusions and future work

From this preliminary work, a HW/SW co-design accelerated QoT estimation tool is demonstrated, enabling an IA CP for dynamic optical networks. A considerable performance improvement is achieved, although it still has substantial improvement margin that can be obtained through the utilization of the hardware floating point unit rather than the software emulated instruction set, which was used for this work.

Acknowledgements

The work described in this paper was carried out with the support of the BONE-project ("Building the Future Optical Network in Europe") and DICONET-project ("Dynamic Impairment Constraint Networking for Transparent Mesh Optical Networks"), funded by the European Commission through the 7th ICT-Framework Programme.

References

1. S. Azodolmolky *et al.* "A dynamic impairment aware networking solution for transparent mesh optical networks," *IEEE Com. Mag*, vol. 47, pp. 38 – 47, 2009.
2. K. Christodoulopoulos *et al.* "A comparison of offline IA-RWA approaches (invited)," in *Proc. of NOC*, 2009.
3. H. Wang *et al.* "Design of a high-performance RSVP-TE signaling hardware accelerator," *IEEE JSAC*, vol. 23, pp. 1588 – 1595, 2005.
4. GNU gprof, http://www.cs.utah.edu/dept/old/texinfo/as/gprof_toc.html.