Demonstration of C/S based Hardware Accelerated QoT Estimation Tool in Dynamic Impairment-Aware Optical Network

Yixuan Qin[(1)], K. C. S. Cheng[(1)], Joan Triay[(2)], Eduard Escalona[(1)], Georgios S. Zervas[(1)], George Zarris[(1)], Norberto Amaya-Gonzalez[(1)], Cristina Cervelló-Pastor[(2)], Reza Nejabati[(1)] and Dimitra Simeonidou[(1)]

[(1)] School of CSEE, University of Essex, Wivenhoe Park, Colchester, UK, ✉ yqin@essex.ac.uk
[(2)] Dept. of Telematics Engineering, Universitat Politècnica de Catalunya (UPC), Castelldefels, Spain

Abstract *An enhanced version of hardware accelerated QoT estimation tool in impairment-aware optical network is demonstrated and evaluated against the different number of lightpaths and wavelengths per link. It outperforms the software version by 28 times in the order of millisecond.*

Introduction

The emerging bandwidth intensive network based applications such as cloud computing unprecedentedly demand high bandwidth optical network with stringent quality of service (QoS) requirements. However, the optical networks suffer from physical layer impairments (PLI). Thus, a legacy service in the IP network may fail on the optical network since the optical signal quality of transmission (QoT) is much more sensitive to PLI. To guarantee an end-to-end transmission service over the optical network, an impairment-aware (IA) optical control plane (CP) is proposed and designed in the DICONET project[1]. The CP deploys a proprietary network planning and operation module (NPOM) to perform a QoT estimation of new, as well as existing lightpaths, upon a connection request. However, the complex algorithmic calculations performed for QoT estimation dramatically increases the overall lightpath setup time, making it unsuitable for real time dynamic lightpath provisioning. Typically, the implementation of the QoT estimation tool is purely software based, which is reasonable for the preliminary module development stage. Based on the routing and wavelength assignment (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 the network load and RWA algorithms deployed. To overcome the long time computation delay in scalable optical networks, a hardware-accelerated QoT estimation tool was proposed and evaluated. Preliminary results were reported in a previous work[3].

In this paper, a more advanced hardware acceleration architecture is proposed, implemented and experimentally demonstrated in an IA dynamic optical network with realtime PLI monitoring. Significant improvement in QoT estimation tool performance is achieved and measured. Meanwhile a client/server (C/S) model between the QoT estimation tool and NPOM is proposed and demonstrated, which offers NPOM a universal hardware acceleration platform, not only for QoT estimation but also for other computation intensive operations, i.e. IA-RWA. Our proposal dramatically 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.

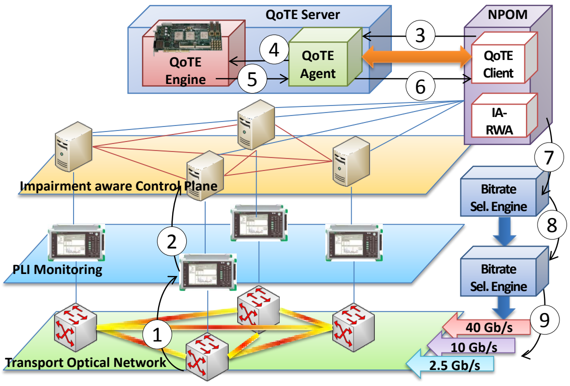

Fig. 1: Architecture of C/S Model Based QoT Estimation Tool in a multi-bitrate optical network with realtime PLI information monitoring.

Architecture of C/S Model Based QoT Estimation Tool

As shown in Fig. 1, the hardware accelerated QoT estimation tool serves the NPOM and replies back with the QoT estimation results. In order to make the implementation of the QoT estimation algorithm independent from the NPOM and to enable performance improvements from task-specific operation, a C/S model is devised. In this case, the *client* is referred as the QoT Estimation Client and resides close to the NPOM, from which it receives lightpath QoT estimation requests, while the *server* consists of the QoT Estimation Engine (QoTEE) (deployed in a Xilinx Virtex IV field programmable gate array (FPGA)) and the QoT Estimation Agent (QoTEA) (running on a 300MHz IBM PowerPC 405 hard core embedded inside

978-1-4244-8535-2/10/$26.00 ©2010 IEEE

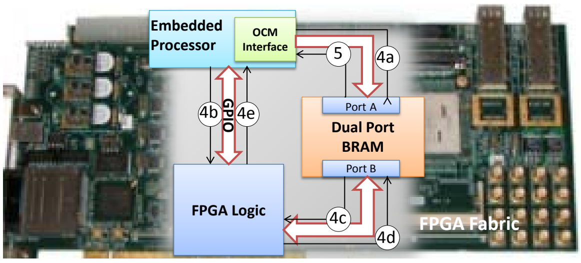

Fig. 2: QoT estimation Server Hardware Part Implementation

the FPGA fabric, in conjunction with 1GB DDR2 memory). The modularity of the QoT server firstly, improves the performance for specific intensive applications, and secondly, enables the reusability of the server implementation for other network planning operations.

The QoTEC monitors realtime PLI measurements (OSNR and optical power measurements in this paper) captured from PLI monitors (steps (1) and (2) in Fig. 1) and updates the QoT estimation tool inputs with the latest PLI information. Upon the NPOM's lightpath request (or a set of lightpaths), the QoTEC generates QoT estimation tool inputs which mainly consist of five data types lightpath routing, lightpath wavelength usage, transmission and reception data and, finally, the data of the physical components that conform the link spans. All this information is transmitted through a socket connection to the QoTEA (3), who will do the initial processing of the received data and present to QoTEE in a suitable format for hardware processing (4). After making the QoT estimation inside the QoTEE, the results are delivered back to NPOM (5 and 6) over the same network connection. An interesting feature of the present architecture is the use of a Bitrate Selection Engine connected to NPOM (7) and capable of deciding to reduce to lower bit rate in order to meet the QoT constraints from the Q values calculated on previous steps (8) and (9). The description of this part is out of the scope of this paper due to lack of space.

Hardware Accelerated QoT Implementation

Fig. 2 shows the hardware part of the QoT estimation tool. It performs most of the QoT estimation calculation and consequently enhances the acceleration in the FPGA. In this paper, a novel hardware acceleration architecture compared to the previous work[3] is proposed, implemented and experimental evaluated. The new architecture uses advanced resources of the state of the art FPGA and its embedded processor, and On Chip Memory (OCM) based interface between the embedded processor and FPGA logics. As a result, a great improvement in acceleration is obtained.

The nature of QoT estimation mathematical algorithm is successive, in other words, the next calculation step is always strictly depended on the previous step. This would cause two major problems when using an APU based interface (in previous work). APU introduces extra overhead when decoding the extended instructions, sending operands to FPGA logic and getting back the results then forwarding them to embedded processor for every extended instruction. In our case, the overhead can accumulate to distinct level while the network size increases. The second problem is that the interrupt when waiting APU will break the hardware pipeline which is the key technique broadly used by hardware acceleration applications. The new OCM interface based approach approach proposed in this paper holds all the estimation algorithms inputs data in a dual port block ram (DPBRAM) and gives the FPGA logic the opportunity to access the input data without any interrupt waiting time, and consequently smoothing the pipelining.

The working flow between the QoTES software part and the hardware part is as follows (see Fig. 2): first, the QoTEA running on the FPGA embedded processor stores the input data from the QoTEC in DPBRAM (4a) following a word-aligned data format recognisable by the hardware algorithm logic. This feature is enabled by memory mapping the physical DPBRAM addresses onto the operating system (OS) running in the embedded processor. Then, the (4b) software program explicitly triggers the start signal in order to initialise the computation of the Q factor. In this process, the hardware retrieves the data from the DPBRAM (4c) to compute the set of fields taken into account by the QoT algorithm. Meanwhile, the software part waits for the end interruption signal eventually set by the hardware when the Q factor computation finishes. When this signal is asserted (4e), it means that the Q factor values are available in the memory, and the server software part can retrieve them (5). Once the values are available, these are sent back to the QoT client program to decide whether the establishment or the proposed lightpaths requests still ensures the minimum quality requirements.

Experimental Demonstration Results and Performance Study

The QoT estimation tool is evaluated realtime in a testbed network environment as shown in in Fig. 1. This testbed has 32 lightpaths and 8 wavelengths per link. In order to evaluate the QoT estimation tool performance against different network scenarios in terms of number of lightpath (LP={5,10,25,50,100}) established in the network and number of wavelength per link, varying from 8 to 32 with a step of 8, and for the scalability, an offline experiment is carried out as well. The non-accelerated QoT estimation tool is evaluated in a very fast general purpose computer which is 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.

Given the above scenarios, Fig. 3 (a) shows the estimation tool performance for a set of number of lightpath 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 28 factors in the order of milliseconds at the worst case 100 lightpaths. According to the trend of the curve, for more lightpath (200, 400 lightpaths), the acceleration performance is expected to be better. This benefits from the hardware pipeline and parallelism.

Fig. 3 (b) shows the QoT estimation tool performance for 100 existing lightpaths and variable number of wavelength per link. The non-accelerated QoT estimation tool execution time is ranging from 18.3s to 27.28s; the corresponding accelerated computation is constantly 0.97s, again below 1 second. The performance improvement is more than 28 times. The QoT Estimation Tool includes single channel effects (ASE noise, PMD, SPM and chromatic dispersion) and multi-channel effects (XPM and FWM), such that in theory all of number of channels and number of established lightpaths impact on its running time. However, in this enhanced hardware accelerated version, the executing time is not associated with the number of wavelength per link. The reason is that in this hardware implementation, the functions in the QoT estimation algorithm which take the number of wavelength per link as a parameter run in parallel along with the rest functions which do not take it as a parameter and have the longest executing time (longest delay line), and the final executing time is determined by the longest delay line which is not related to the number of wavelength per link. This gives hardware accelerated QoT estimation tool extra scalability.

Conclusion

An novel HW/SW co-design accelerated QoT estimation tool is demonstrated, enabling an IA CP for dynamic optical networks. A significant performance improvement is achieved for control plane. The QoT estimation tool executing time is in the order of millisecond for the worst case (100 ligthpaths) and remains constant as number of wavelength per link increases.

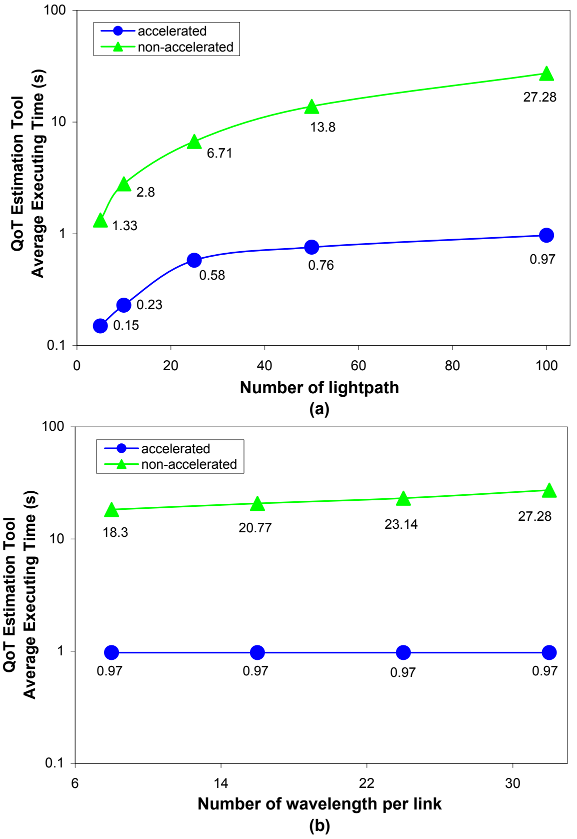

Fig. 3: QoT estimation tool average executing time *v.s.* (a) the number of lighpaths for 32 wavelenfths per link and (b)the number of wavelength per link for 100 lightpaths

Acknowledgements

The work described in this paper was carried out with the support of the 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. *Communications Magazine, IEEE*, 47(5):38–47, May 2009.

2 K. Christodoulopoulos, *et al.* A comparison of offline ia-rwa approaches (invited). In *Proc. of NOC 2009*.

3 Y. Qin, *et al.* Hardware accelerated impairment aware control plane. In *Proc. OFC 2010*.