Software Defined Optical Switching for Cloud Computing Transport Systems

Taras Maksymyuk
Korea University, Department of Computer and Information Science,
2511 Sejong-ro, Sejong City 339-770, S. Korea
+82-044-860-1755
taras.maksymyuk@gmail.com

Stepan Dumych
Lviv Polytechnic National University, Department of Telecommunications
12 S. Bandery str., Lviv 79013, Ukraine
+38-032-258-2444
stiv.dumych@gmail.com

Olena Krasko
Lviv Polytechnic National University, Department of Telecommunications
12 S. Bandery str., Lviv 79013, Ukraine
+38-032-258-2444
krasko.lena@gmail.com

Minho Jo
Korea University, Department of Computer and Information Science,
2511 Sejong-ro, Sejong City 339-770, S. Korea
+82-044-860-1348
minhojo@korea.ac.kr

ABSTRACT
This paper presents a new approach for optical burst switching in order to achieve better performance of cloud computing systems. The model of software defined burst scheduling in core node of OBS (Optical Burst Switching) network was designed based on field programmable gate arrays supervised by main controller. Simulation results prove the advantage of the proposed network infrastructure.

Categories and Subject Descriptors
C.2.5 [**Local and Wide-Area Networks**]: High speed networks – *fiber channel, optical switching, software defined networks.*

General Terms
Algorithms, Performance, Design.

Keywords
Optical Burst Switching, Burst Header Packet, Software defined bursts scheduling.

1. INTRODUCTION
Current development of telecommunication networks brings a lot of new opportunities for users such as cloud computing, multimedia streaming and real-time interactive applications [1]. Modern research on this topic has been devoted to designing new methods for distributed computing and data transmission between highly distant data centers. The performance of any distributed cloud system and quality of service provision depends on different factors. In particular, the physical throughput between remote data centers significantly influences on distributed cloud computing performance. But distributed cloud computing requires mechanisms of data integrity and security provision on the link layer. On the other hand, cloud systems should support specific load balancing and resource management algorithms combined with reservation and quality of service maintenance algorithms, which need full convergence of different communication standards. Thus, on the transport layer, network is considered as a set of virtual channels with adaptive throughputs, allocated for support of some separate services [2]. In this paper we propose a software defined approach for optical burst switching networks [3]. The proposed approach is designed to increase the performance of cloud services provision by adjusting transmission modes according to given requirements. We develop a software defined burst scheduling model based on field programmable gate arrays in order to reduce burst offset time and achieve better network performance. This paper is organized as follows. Section II covers the OBS system model and proposed software defined bursts scheduling algorithm. Section III concludes the proposed work.

IMCOM '15, January 08 - 10 2015, BALI, Indonesia

http://dx.doi.org/10.1145/2701126.2701232

2. SOFTWARE DEFINED OBS NETWORK MODEL
We consider two typical scenarios of interaction in distributed cloud systems via OBS network: virtual machine migration and joint service provision. The main difference between them lies in channel requirements. Virtual machine migration is usually carried out ahead of a service request in destination cloud. Therefore, the virtual machine migration does not require burst delivery time. However, this scenario strictly requires data integrity in order to create exact copies of service instances in destination cloud. Therefore, "tell and wait" signaling scheme with confirmation of established channel is used and large burst size is preferred in order to decrease quantity of burst header packets (BHP) [4]. For joint service provision the main requirement is very fast data processing and transmission in order to maintain SLA (Service Level Agreement) policy for cloud users. Therefore, "just enough time" (JET) signaling scheme, without channel confirmation, is used [4] and small burst size is preferred due to limitation of packets latency. The architecture of Software Defined OBS network is shown in Fig.1. According to given network circumstances, the controller computes all requested transmissions within its domain and sends the control data to edge nodes. In our system model each edge node is connected by n channels and each channel contains m wavelengths. Controller sends a [n×m] parameters matrix to each edge node by BHP.

We define matrix **A** as burst destination address, matrix **C** as burst priority according to service quality requirements. **T** is the burst offset time and **BS** is the burst size. We propose new algorithm of software defined burst scheduling in core nodes by using field programmable gate arrays.

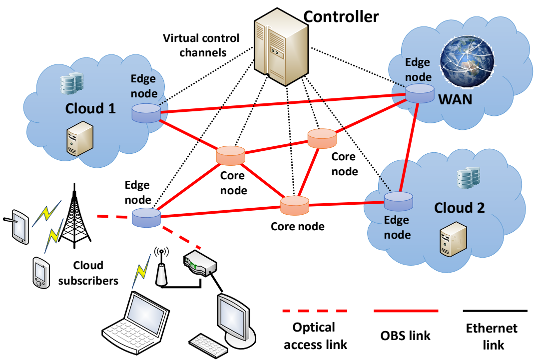

Figure 1. Architecture of Software Defined OBS network

Proposed algorithm is described below.

Step 1. Core node gets the following information about burst from received BHP: size – BS_{ij}, wavelength – *j,* channel – *i*, destination node address –A_{ij}, burst priority – C_{ij}, offset time – T_{ij}.

Step 2. By using routing protocols specified by the network controller, a set *z* of available routes for the destination node A_{ij} is represented by the vector:

$$VC_{ij} = \left[VC_{ij1} \quad \cdots \quad VC_{ijk} \quad \cdots \quad VC_{ijz}\right]^T \qquad (1)$$

where VC_{ijk} is one of the routes to the destination.

Step 3. From the set (1) the preferred route for burst transmission is selected according to the following condition:

$$VC_{ijk} = \begin{cases} \min\left(VC_{ij}\{\tau_\Sigma\}\right), 8 \le C_{ij} \le 15 \\ \forall \in \left[VC_{ij}\right], 0 \le C_{ij} \le 7 \end{cases}, \left(0 < i \le n, 0 < j \le m, 0 < k \le z\right) \; (2)$$

where τ_Σ – route total latency, including T_{ij}, $0 \le C_{ij} \le 7$ denotes the burst priority for virtual machine migration scenario and $8 \le C_{ij} \le 15$ is for joint service provision. Thus, the total virtual channels matrix **VC** is combined containing information on a proper route for each burst. If traffic intensity is not very high, controller will proceed with the algorithm to the step 9. In case of high traffic intensity the algorithms will proceed to step 4.

Step 4. In order to adjust each virtual channel more precisely towards quality of service requirements of specified burst, the elements in vector (1) are sorted in descending order as following:

$$\left(\overrightarrow{VC}_{ij}\right)_k = \max\left(VC_{ij}\{\tau_\Sigma\}\right), \left(VC_{ij} \in [k, z], k = 1, 2 \ldots z\right) \qquad (3)$$

Step 5. From address matrix **A** the set of binary equal destination matrices is formed into:

$$D_{sl}^{\{A_{ij}\}} = \begin{cases} 1, A_{sl} = A_{ij} \\ 0, A_{sl} \ne A_{ij} \end{cases}, \left(0 < s \le n, 0 < l \le m\right) \qquad (4)$$

Matrix (4) shows the indices of bursts with the same destination address.

Step 6. For bursts with equal destination, the reduced priority matrix will be calculated as following:

$$P_{ij}^{\{A_{ij}\}} = C_{ij} D_{ij}^{\{A_{ij}\}}, \left(0 < i \le n, 0 < j \le m\right) \qquad (5)$$

Step 7. Matrix (5) is transformed to the column-vector as following:

$$\vec{P}_k = P_{ij}^{\{A_{ij}\}}, \vec{i}_k = i, \vec{j}_k = j, \; P_{ij}^{\{A_{ij}\}} \ne 0, \; 0 < i \le n, \; 0 < j \le m, \; 0 < k \le z \quad (6)$$

where i_k, j_k are column vectors of income bursts indices with the same destination address, P_k is column vector of income bursts priorities with the same destination address.

Step 8. In order to allocate proper channel for each burst, vectors (6) sorted in ascending order for achieving indices matching between priority and corresponding channel quality:

$$\vec{P}_k = \min\left(\vec{P}_\zeta\right), \; \vec{i}_k = \min\left(\vec{i}_\zeta\right), \; \vec{j}_k = \min\left(\vec{j}_\zeta\right), \; \zeta \in [k, z], 0 < k \le z \; (7)$$

Thus, we obtain the result of switching calculation according to the following rule:

For income burst with wavelength $\mathbf{j_k}$*, within channel* $\mathbf{i_k}$*, priority* $\mathbf{P_k}$*, and destination address* $\mathbf{A_{ij}}$*, the optimal virtual channel is* $\mathbf{VC_k}$*, i.e., the virtual channel with lowest latency will be allocated for bursts with highest priority.*

Step 9. Bursts are forwarding towards destination nodes.

We have simulated the performance of proposed software defined burst scheduling algorithm (SDBS) comparing to recent linear adaptive scheduling algorithm (LAGS) [5]. Fig.2 presents the CDF of burst scheduling time for both algorithms. You can notice that our proposed algorithm SDBS is obviously superior to the existing LAGS algorithm.

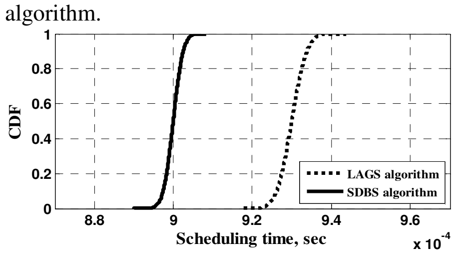

Figure 2. Cumulative distribution functions for burst scheduling algorithms

3. CONCLUSION

We propose a new software defined approach for OBS network planning and operating in cloud systems. The proposed software defined burst scheduling algorithm SDBS is designed to support the proposed architecture. Simulation results show the advantage of SDBC algorithm in terms of scheduling time.

4. REFERENCES

[1] He, J. 2013. Software-defined transport network for cloud computing. *In Optical Fiber Communication Conference/National Fiber Optic Engineers Conference*, (Anaheim, California, United States, March 17-21, 2013), paper OTh1H.6.

[2] Dumych, S., Maksymyuk, T., Krasko, O., and Guskov, P. 2013. The Virtual Channel Parameters Calculation in All-Optical Network. *IEEE International Conference on The Experience of Designing and Applications of CAD Systems in Microelectronics* (Polyana-Svaliava, Ukraine, Feb.2013), p.87.

[3] Qiao, C., and Yoo, M. 1999. Optical burst switching (OBS)—A new paradigm for an optical internet. *J. High Speed Networks*, vol. 8, pp. 69–84..

[4] Dumych, S., Guskov, P., Maksymyuk, T., and Klymash, M. 2013. Simulation of characteristics of optical burst switched networks. IEEE International Conference on Microwave & Telecommunication technology, (Sevastopol, Ukraine, Sep. 2013), pp. 492-493.

[5] Nhat, V. V. M., and Quoc, N. H. 2014. A model of adaptive grouping scheduling in OBS core nodes. Journal of Convergence, vol. 5, no.1, pp. 9–13.