Design and Implementation of an Adaptive Assembly Algorithm for Edge Nodes of Satellite OBS Networks

Lisiyang Li¹, Lufeng Qiao*¹ and Qinghua Chen¹

¹ Institute of Communication Engineering, Army Engineering University of PLA, Nanjing Jiangsu 210007, China.
*Corresponding author’s e-mail: 13357837783@189.cn

Abstract. The edge nodes of the satellite optical burst switching network are the interface devices of the ground network and the satellite optical burst network. Ground network data needs to be classified, assembled, and generated bursts at the edge nodes. The burst will be transmitted on different wavelength channels. Reasonable selection of the assembly algorithm of the edge nodes can reduce the design complexity and improve the forwarding performance and QoS capability of the satellite optical burst switching network. In this paper, an adaptive assembly algorithm is proposed to adjust the length threshold and time threshold dynamically. The basic principle is to evaluate current network traffic depend on the actual burst parameters of previous bursts, and then determine the specific value of length threshold and time threshold. The algorithm is implemented on a Xilinx's XC7VX690T-2FFG1927I FPGA, and verified by Modelsim SE 10.6d.

1. Introduction
For the past few years, new demands have been put forward for the processing capacity of satellite network nodes [1] with the rapid development of the increasing application of satellite and inter-satellite laser links, and people's increasing requirements for high efficiency, large capacity networks. With the increasing demand for satellite network capacity, how to solve the problem of electronic bottleneck becomes more and more important [2]. Compared with the traditional electrical on-board switches, optical switching technology can provide a larger exchange capacity, which is attractive to researchers in recent years.

Traditional optical switching technology can be divided into optical circuit switching (OCS) [3], optical packet switching (OPS) [4] and optical burst switching (OBS) [5]. OBS takes advantage of coarse-grained OCS and fine-grain OPS. Its requirements for optical logic devices are not critical, and the switch fabric structure is relatively simple. This paper applies the mature ground optical burst technology to the satellite switching network and proposes an adaptive assembly algorithm for the edge node of the satellite OBS network.

2. Architecture of OBS satellite network
The satellite OBS network refers to the basic principles of terrestrial optical burst switching networks. It divides the entire network architecture into satellite core switches, satellite edge gateways, laser links, microwave links, other network resources and terrestrial networks. Figure 1 shows the overall architecture of the satellite OBS network.
The satellite edge gateway, as the edge node of the entire satellite OBS network, first classifies the data services from different networks according to different destination ports and quality of service (QoS), and then generates bursts according to different assembly algorithms. The key information of the BDP, such as burst length, offset time, QoS information, destination port address, is stored in a Burst Control Packet (BCP) which is sent to the core switch to reserve resource. In the whole process, BCP uses one control channel alone, and BDP uses several data channels. After receiving the BCP, the core switch extracts the information from the BCP by optical-electrical conversion, and reserves the corresponding resources for the BDP. After a certain offset time, the edge node sends a BDP that is much larger than a traditional IP packet to the core switch. After the BDP is at the core switch, the core switch has already established an optical channel for it. The BDP will be directly forwarded through this wavelength channel without any processes. This forwarding process increases the transmission rate and eliminates the "electronic bottleneck" problem greatly.

3. Design of assembly algorithm
In recent years, most research has focused on time thresholds [7], length thresholds [8], time-length mixing thresholds [9], and thresholds with adaptive feedback assembly strategies [10]. While these methods, are not suitable for short flows, which will cause the traffic change greatly during a short time. This design proposes an adaptive assembly algorithm that can dynamically adjust the assembly thresholds. The algorithm can improve the utilization of channel bandwidth by recording the actual value of two previous burst parameters, like burst length and time used to generate the burst, then calculate the dynamic adjustment length threshold and time threshold. Figure 2 is the flow chart of a length-time adaptive assembly algorithm.
When the burst assembly circuit module is initialized, the maximum BDP length value \( L_{\text{max}} \), the minimum BDP length value \( L_{\text{min}} \), the maximum time value \( T_{\text{max}} \) and the minimum time value \( T_{\text{min}} \) are set. The initial length threshold is \( L_0=(L_{\text{max}}+L_{\text{min}})/2 \), the initial time threshold is \( T_0=(T_{\text{max}}+T_{\text{min}})/2 \), the length threshold adjustment parameter is \( L_a=(L_{\text{max}}-L_{\text{min}})/n \) and the time threshold adjustment parameter is \( T_a=(T_{\text{max}}-T_{\text{min}})/m \) (n and m are parameters used to change the length and time thresholds during dynamic adjustment). Parameters of m and n can be configured by embedded processors based on the burst priority and the network performance during a period of time. At the same time, a counter cnt and \( c_{\text{max}} \) are set for dynamic adjustment. After a data packet enters the burst assembly circuit, it will check if the timer reaches \( T_0 \) first, if the timer has not reached \( T_0 \), it will then check if \( L_0 \) has been reached. If \( T_0 \) or \( L_0 \) has been reached, a BDP generation command is generated to form a BDP and at the same time the statistical module will record current threshold value. Each current assembly threshold will be compared with the previous one. If the same assembly threshold is used twice, the value of cnt will be add 1. If two different assembly thresholds are used, cnt will be cleared. When cnt reaches \( c_{\text{max}} \), it is realized that the last assembly thresholds are same. If both times are \( T_0 \), the length is not reached, it is realized that there are more gaps in the data flow. So set \( L_0 \) to \( L_0+L_a \), \( T_0 \) to \( T_0+T_a \). If both are \( L_0 \) triggers, indicating that they can quickly assemble to the length threshold in a period of time without triggering the time threshold, so set \( L_0 \) to \( L_0 \) \( L_a \). With the purpose of reducing the number of BCPs, the length threshold will be add a \( L_a \). When both thresholds are triggered irregularly, the threshold is set to the initial value. This assembly algorithm aims to dynamically modify the trigger threshold, reduce the interval between bursts, and improve channel utilization by adaptively changing the values of the length threshold and the time threshold in such an irregular network.

4. Main simulation results
Figure 3 shows the burst assembly circuit adjusts the length threshold dynamically. The test program continuously sends test data to simulate a high network load. The \( \text{wr}_\text{state} \) is the BCP command generation state machine for the burst assembly circuit, and the \( \text{cros}2\text{ddr}_\text{data}_\text{sfifo}_\text{din} \) is the data which is written to DDR3 from the pre-stage circuit. Firstly, the data is cached in the FIFO and then written in the DDR3. The burst command generation module updates the burst length in real time according to the size of the buffer and generates a burst command \( \text{rd}_\text{cmd}_\text{din} \) (128 bits include the
read address from DDR3, the number of cells to be read, the destination address of the burst, the burst address and QoS information). A burst generation command is formed by comparing the number of cells in the buffer with the length threshold. When the same length threshold is used several times and the length threshold is consistent, indicating that the network load is high, the length threshold should be appropriately increased. In Figure 6, the length threshold length_th is set to 200. When the adjustment condition is met, length_th will be add 10 each time.

Figure 3. Simulated waveform of adjusting the length threshold dynamically.

Figure 4 shows the burst assembly circuit adjusts the time threshold dynamically. The test program sends data at intervals, where cros2ddr_data_sffo_din is the data generated by the pre-stage module and ready to be sent to DDR3. In figure 4, there is some idle time in the middle, and the test program sets the time interval between each frame to simulate a situation when the network load is low. The current length threshold is 200 cells and the time threshold is set to 400 cycles. So in a full load situation, one time threshold corresponds to a data volume of 400 cells when read a cell needs one period. Firstly, the burst assembly circuit records the threshold value of the data trigger twice. Secondly, if the two times are triggered by the time threshold, indicating that the length threshold is not reached, and the amount of data in 400 periods is less than 200 cells, so this time threshold should be reduced. If the time threshold has been reduced to the T_{min} and the number of burst packet cells has not changed in the last few times, the time threshold will be kept at T_{min}. If the data amount changes, the time threshold will be set to the initial value. In the figure 4, wr_state is the state machine that generates the BDP generation command by the assembly module, rd_cmd_din is the BDP generation command, time_th is the time threshold, and time_th_1 is the time consumed by the burst assembly in the current queue. The initial value of time_th is set to 400, which is automatically adjusted to 390 to reduce the consumption of excess time and increase channel utilization. The dynamic adjustment value here is set to 10, and it can also be flexibly configured through the register according to customer needs.

Figure 4. Simulated waveform of adjusting the time threshold dynamically.

5. Conclusion
In this paper, an adaptive assembly algorithm is designed for the edge node of satellite optical burst switching network. In theory, the burst interval can be reduced, the channel utilization rate can be increased, and the overall network performance can be improved. The algorithm is implemented by XC7VX690T-2FFG1927I FPGA and verified by Modelsim SE 10.6d, which can meet the requirements of satellite OBS network.
References
[1] Ma, J. (2015) Satellite optical communication. National Defense Industry Press, China.
[2] Yuan, J. Wang, Y. Meng, X. (2006) Bent pipe satellite network system and area routing method. J. Space Sci, 26(4):315-320.
[3] Regis, J. (2001) Optical Switching and Networking Handbook. McGraw-Hill Publisher. New York.
[4] O’MAHONY, M. J. (2001) The application of optical packet switching in future communication networks. J. IEEE Communication Magazine, 39(3): 128-135.
[5] Chen, Y. Qiao, C. Xiang, Yu. (2004) Optical burst switching: a new area in optical networking research. J. Journal of IEEE Network, 2004, 18(3): 16-23.
[6] Le, Z. C. Zhu, Z. J. (2015) Optical burst switching technology, China Machine Press, China.
[7] GE, A. CALLEGATI, F. TAMIL, L. (2000) On optical burst switching and self-similar traffic. J. IEEE Communication Letters, 4(3): 98-100.
[8] OH, S. KANG, M. (2002) A burst assembly algorithm in optical burst switching networks. In: Proc.of Optical Fiber Communication Conference and Exhibit, Anaheim, USA, 771-773.
[9] KANTARCI, B. OKTUG, S. ATMACA, T. (2006) Performance of Optical Burst Switching techniques under self-similar and poisson traffic based on various burst assembly techniques. J. Computer Communication, 30(2): 315-325.
[10] Xia, H. Z. Yuan, B. L. (2016) Research on self-similarity traffic assembly algorithm in OBS networks. J. Optical Communication Technology, 40(1): 44-46.