OBS边缘节点中突发包组装算法的实现

周 纪[1]，程 亮[2]，胡 钢[1]

(1. 电子科技大学 宽带光纤传输与通信网技术教育部重点实验室 成都 610054; 2. 青岛有线电视网络中心 山东 青岛 266071)

【摘要】研究了最小长度最大汇聚时间算法在现场可编程门阵列中的实现方法。在实现过程中，使用了状态机控制组装流程。在缓存的使用上，提出了共享缓存的做法以节省存储资源，即根据输入数据流的随机性，实际设置的汇聚缓存数目小于总的突发包类型数目。

关 键 词 光突发交换; 边缘节点; 突发队列; 共享缓存

中图分类号 TN929.11 文献标识码 A

Implementing Burst Assembly in OBS Networks Edge Node

Zhou Ji[1]，Cheng Liang[2]，Hu Gang[1]

(1. Key Laboratory of Brodband Optical Fiber Transmission and Communication Networks UEST of China, Ministry of Education Chengdu 610054;
2. Qingdao CATV Shandong Qingdao 266071)

Abstract How to assemble TCP/IP packets into data burst in OBS edge nodes is discussed. Min-burst-length-max-assembly-period algorithm is preferred and implemented on Xilinx® VirtexII Pro FPGA. A state machine controls the assembly process. Sharing buffer memory is proposed to save memory resources. That is, the number of physical burst queues is less than that of burst types since the inputs is random.

Key words OBS; edge node; burst queue; sharing buffer memory

光突发交换(Optical Burst Switching, OBS)是近年来提出的一种新技术，它使用的带宽粒度介于光电路交换和光分组交换之间，在独立的信道上传送控制分组，通过控制分组携带的信息在经过的核心节点上为突发包预留资源，突发包则在光域以直通的形式通过整个OBS网络。核心节点只需对控制分组进行E/O/E转换和处理，这样就克服了光电路交换中的交换瓶颈，提高了带宽利用率，且比光分组交换更易于实现，可以说是两者之间的平衡选择。

突发包的组装是OBS的一项关键技术，直接影响到OBS的性能。它的核心是组装算法的选择和实现。文献[1]中提出了几种算法，并比较了它们各自的优缺点，其中算法C——最小突发长度最大汇聚时间(Min-Burst-Max-Assembly-Period, MBMAP)——兼顾了包长和组装时间两个参数，既能有效地利用网络带宽，又避免了大的网络延迟，是一种较为优越的算法。

1 突发包格式

突发包是OBS网络中数据传输的基本单元，它是由一些具有相同属性(如QoS等级、目的节点地址等)的IP分组组成的[2]。每个突发包配有一个控制分组(Burst Control Packet, BCP)，它先于突发包发送，其中携带

收稿日期：2004 – 07 – 09
基金项目：国家863计划资助项目(2002AA122021)
作者简介：周 纪(1980 –)，男，硕士生，主要从事光突发交换网络方面的研究.

了突发包的一些信息(如长度、目的边缘节点地址等)，用来在经过的核心节点为突发包预留资源，经过一个偏置时间后再发送突发包，有效载荷就能以直通的形式在光域传输，从而减小了核心节点E/O/E转换和电域处理的负担。

关于突发包的格式，目前还没有统一的标准，本文中所讨论的格式如图1所示。

BT	NOP	BL	Payload

图1 突发包格式

图中BT为突发包类型，NOP为突发包中包含的IP分组数， Payload为净荷，BL为净荷长度。突发包类型按照其中IP分组的QoS等级和目的边缘节点的地址分类。设有M种优先级，N个边缘节点，那么对任意一个边缘节点中形成的突发包，其目的边缘节点地址都有$N-1$种可能，于是在每个边缘节点中组装的突发包共有$M\times(N-1)$种。

2 突发包的汇聚和组装

MBMAP算法的基本思想是，当累积IP分组的长度超过设定的最小突发长度MBL或者汇聚时间超过时，发出一个控制分组，经偏置时间offset time后，发送突发包。下面讨论该算法的实现方法。

2.1 共享缓存

按照前面的假定，每个边缘节点都有$M\times(N-1)$种类型的突发包，如果为每种类型的突发包单独分配一个缓存，那么随着M与N的增大，需要的缓存容量将会急剧增加；另一方面，由于到达边缘节点的IP分组类型的随机性，任意时刻都可能有部分缓存闲置，而另一部分缓存则因负担过重而有可能丢失数据。这样一来，既降低了网络的性能，又浪费了大量的资源。

为了改善网络性能并提高缓存资源的利用率，可以采用共享缓存的办法[3]。基本思想是，将每个边缘节点的缓存数目配置为K个(其中K为小于$M\times(N-1)$的正整数)，当有一个IP分组到达时，交换矩阵根据其类型和当前各缓存的状态决定将其发往哪个缓存，如图2所示。线卡是OBS边缘节点与外部以太网的接口。交换矩阵前配置缓存是为了便于交换矩阵识别IP分组的头信息和实现线卡与汇聚缓存之间的速率匹配。

采用动态缓存的做法需要为每个汇聚缓存打上一个动态的标记，用以表明每个缓存当前的状态是空或是有某种类型的突发包正在该缓存中组装。当有一个IP分组到达时，交换矩阵轮询当前各汇聚缓存的状态，若发现某个缓存中待组装的突发队列与当前IP分组类型一致时，就将分组送到该队列中；若没有发现相匹配的突发队列，则将该IP分组送入一个空闲缓存中，启动一个新的突发包组装队列，同时将该缓存打上相应的标记。由于交换矩阵前面的缓存与后面的突发汇聚缓存不是一一对应的关系，各时刻同一汇聚缓存中突发包的类型也不确定，所以就加大了交换矩阵和后级发送模块轮询算法的复杂性。

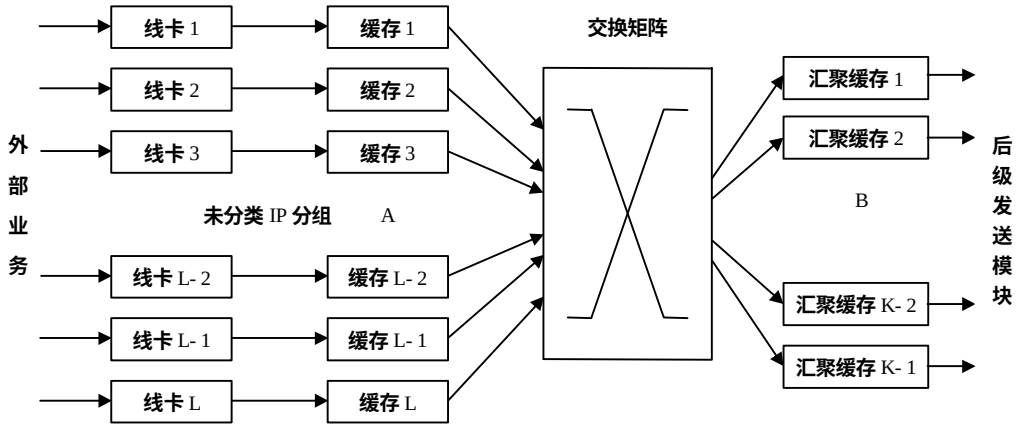

图2 共享缓存实现示意图

2.2 汇聚算法

本文针对MBMAP算法具体实现中的问题对文献[1]中描述的MBMAP算法作了适当的调整。设BL_i为当

前队列中正在组装的第i类突发队列的长度，PL_i为当前到达的IP分组的长度，t_i为计时器的当前值，T为计时门限。当有一个第i种类型的IP分组到达时，组装算法的描述如下：

```
Event::Packet arrive(i)
    if (BL_i = 0) then
        Beginning the timer;
        Set the flag of the queue;
    end if;
    BL_i = BL_i + PL_i;
    if (BL_i > MBL  or  t_i > T) then
        Generate a sending request;
        Modify the flag of the queue;
    end if;
    Assemble the packet to the queue;
```

```
Event::received response(i)
    Read out the burst;
    Clear and stop the timer;
    Clear the flag of the queue;
    BL_i = 0;
Event::time out
    if (no writing presently) then
        Generate a sending request;
        Modify the flag of the queue;
    end if;
```

图3是算法实现的状态转移图。其中各状态的含义为：idle 代表缓存为空；writeIP代表某IP分组正在写入队列中；timerun代表计时器运行中；wait代表等待发送模块响应；con_write代表发出请求后正在往队列中写入最后一个IP分组；wr_rd代表最后一个IP分组未写完时收到发送模块的响应，在写最后一个IP分组的同时读出突发包；readBurst代表读出突发包。

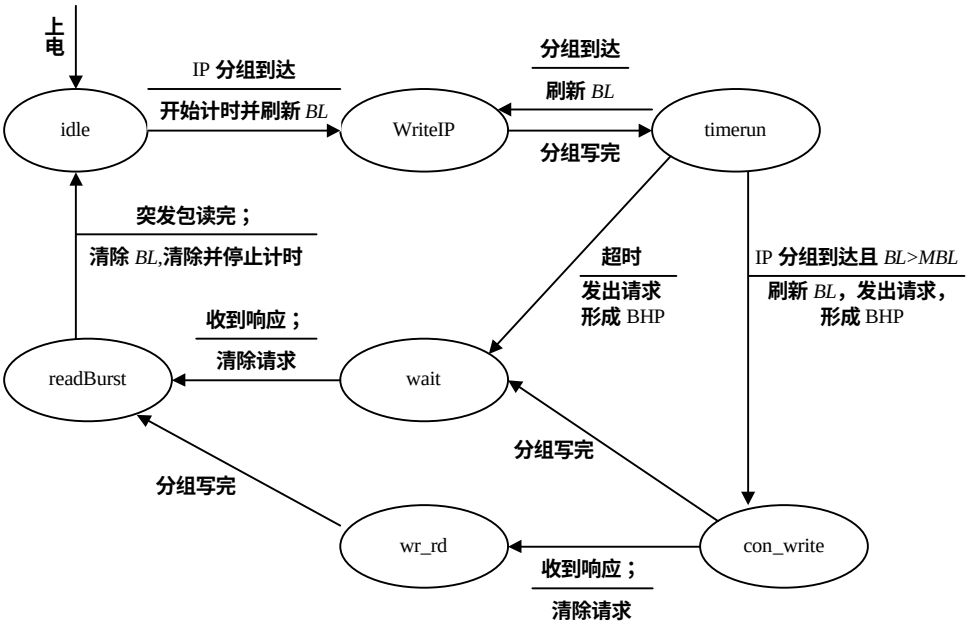

图3　组装算法状态转移图

图中有“分组写完”注解的三处，“分组写完”为转移条件，状态机没有输出或输出维持不变；其余注解处，横线上方的内容为转移条件，下方的内容为状态机输出。上电时的初始状态为idle。图中需要说明的两点是：

1) 向后级发送模块发出的请求并不是最后发送的控制分组，仅仅是包含控制分组所需的信息。发送模块检测到请求时先将这些信息存储起来，待检测到发送端口有空时，根据请求中的信息组装好BHP发送出去，再经过一个偏置时间后读出并发送突发包。因为BHP中需要携带偏置时间信息，所以必须等到能够确定偏置时间值时才能组装好并发送。

2) 由于包头信息中的NOP和BL都是在最后才能确定，为了方便数据读写指针的控制，对应于每一个汇聚缓存，都另外设置一组寄存器，用于存放最后形成的包头。在收到后级发送模块的读信号时，使用一个两路的选择器，先送出包头，紧接着送出汇聚缓存中的净荷(Payload)。

3 现场可编程门阵列实现

设有3个边缘节点，两种QoS等级，突发包组装算法实现涉及的实验中的各项参数如下表。

表1 突发包组装算法实现涉及的实验参数

描述	突发包类型	突发队列	最小突发包长	缓存长度	最大汇聚时间	线卡数
参数	$M\times(N-1)$	K	MBL	$MaxBL$	T	L
参数值	4	3	13 KB	15 KB	80 μs	6

其中每路线卡外接一个千兆以太网。实验系统的核心为Xilinx公司的XC2VP40芯片[4]，软件平台采用Xilinx ise6.1。XC2VP40有8组高速串行数据收发模块，用于以太网数据的接收和突发包的发送；汇聚缓存采用片内Block RAM实现，网络规模较大时可外挂存储器。Block RAM是Xilinx公司系列现场可编程门阵列(Field Programmable Gate Array, FPGA)芯片内经过布局和布线优化的专用存储单元，用户设计中较大的存储模块建议采用它来实现，这样可以提高工作频率。实验系统时钟采用单一的125 MHz晶振。为实现速率匹配，图2中A处缓存采用16位处理，B处汇聚缓存采用32位处理。这样，汇聚缓存中每个单元的宽度为4字节，当IP分组的长度不是4的整数倍时，每个分组占用的最后一个存储单元将有1～3个字节被浪费。但突发包内都是一个个完整的IP分组，所以接收端仍能正确识别数据。

对于本实验中的参数T，由于每路以太网数据经10 B/8 B解码后流向后续模块的数据速率为1 Gb/s，相当于1b/ns，从以太网中接收到的数据被解包后最大的分组为1 500 B，按照本实验所采用方案及表1中的MBL值，每个突发包所能容纳的IP分组数最少约为10个(即10个最大的IP分组)，于是有：

$$T\geq 3\times 10\times 1\,500\times 8/\,6 = 60\,000\ \text{ns} = 60\ \mu\text{s}$$

考虑到业务流的随机性，实验中取T=80 μs。若在某汇聚缓存中的突发队列组装时间刚好超时的那一刻正好有一个IP分组被分发到该缓存，则当前分组应继续汇聚到该队列中，也就是按长度优先的原则，这样可以提高网络带宽的利用率。

当某汇聚缓存发出请求后，应修改它的标记，表明该汇聚缓存中已形成了新的突发包，在该汇聚缓存变为空闲状态以前交换矩阵不要再往它分发新的IP分组，以免造成数据混乱。

实验结果综合后的频率为181.258 MHz、时钟周期约束为6 ns时，实现后工作频率可达176.616 MHz，满足设计要求。每个缓存占用8个Block RAM，也就是8×18 KB的存储空间，可见如不采用共享缓存的方法，需要的存储容量将随网络规模的扩大而以惊人的速度增长。

4 结 束 语

MBMAP算法是目前OBS网络突发包组装算法中较好的一种，它兼顾了包长和组装时间参数，能够较充分地利用网络带宽且不会延误IP分组的发送。本文探讨了该算法在FPGA中的具体实现方法，并提出了共享缓存的做法，在网络规模较大的情况下可显著节省存储资源。

参 考 文 献

[1] Cao Xiaojun, Li Jikai, Chen Yang; *et al*. Assembling TCP/IP packets in optical burst switched networks. [DB/OL]. http://ieeexplore.ieee.org/iel5/8454/26643/01189141.pdf, 2004-07-20

[2] 于金辉，范 戈. 光突发交换中的突发包组装技术研究[J]. 光纤与电缆及其应用技术, 2002, (4): 1 518

[3] 罗洪斌，胡 钢，李乐民. 光突发交换网络边缘节点突发排队方案[J]. 电子科技大学学报，2003，32(3)：289-292

[4] Xilinx®.Virtex-II Pro/ Virtex-II Pro X Complete Data Sheet. [DB/OL]. http://www.xilinx.com/ 2004-06-30

编 辑 熊思亮