A novel scheme based on minimum delay at the edges for optical burst switching networks

Jinhui Yu (于金辉), Yijun Yang (杨毅军), Yuehua Chen (陈月华), and Ge Fan (范 戈)

National Laboratory on Local Fiber-Optic Communication Networks & Advanced Optical Communication Systems, Shanghai Jiaotong University, Shanghai 200030

Received September 16, 2002

This paper proposes a novel scheme based on minimum delay at the edges (MDE) for optical burst switching (OBS) networks. This scheme is designed to overcome the long delay at the edge nodes of OBS networks. The MDE scheme features simultaneous burst assembly, channel scheduling, and pre-transmission of control packet. It also features estimated setup and explicit release (ESXR) signaling protocol. The MDE scheme can minimize the delay at the edge nodes for data packets, and improve the end-to-end latency performance for OBS networks. In addition, comparing with the conventional scheme, the performances of the MDE scheme are analyzed in this paper.

OCIS codes: 060.4510, 060.4250.

Currently, a new switching technology, OBS[1], has been put forward to meet the demand for high-speed IP routers for the rapidly growing Internet. On one hand, OBS is more flexible and has higher bandwidth utilization than circuit switching (CS) because its granularity of switched unit is smaller than that of CS. On the other hand, it is more feasible than optical packet switching (OPS) because it uses out-of-band control signal to exclude optical buffering at core nodes. An OBS network consists of optical core nodes and electronic edge nodes connected by WDM links. In OBS networks, the incoming data-packets which have the same destination(s) (egress edge node) and quality of service (QoS) requirements are assembled into bursts at electronic ingress edge nodes. Bursts are then routed at optical core nodes through the OBS network and disassembled back into packets at egress nodes to be forwarded to the next hops.

The conventional scheme for ingress edge nodes of OBS works in the following order[2]: burst assembly, burst queuing, data channel scheduling, generation and transmission of control packet, and burst transmission. So the total delay for data-packets is long, and does not suit the strict real-time service.

We propose a new scheme based on minimum delay at the edges, called MDE scheme, to solve this problem. The MDE scheme has three characteristics: simultaneous burst assembly and channel scheduling (SAS); pre-transmission of control packet; and ESXR for switching path at the core nodes.

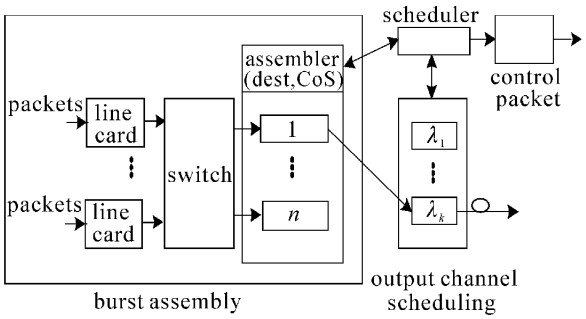

Fig. 1. The block diagram for ingress edge node.

A typical functional architecture of the ingress edge node is shown in Fig. 1. A predefined time threshold, called burst assembly time, controls the burst assembly when the incoming data-packets are switched into the assembler. So a burst will be generated after the assembly time has elapsed, and the burst arrival time is available to the scheduler for the specified outgoing link. Considering this point, we propose the SAS method. SAS works as follow: an assembly timer starts when the first data-packet arrives at the assembler. At the same time, the assembler sends a requirement to the scheduler to schedule outgoing channel for the assembling burst. In addition, the burst arrival time which is equal to current time plus burst assembly time is also send along with the scheduling requirement. The scheduler selects a channel for the burst. Then, a control packet is generated to carry control information which includes source address, destination address, offset time, the outgoing channel number, and QoS requirements for the burst. Denoting the current total length of all data-packets in the assembler by l_b, the value of assembly timer by t_c, the burst assembly time by t_{ass}, and the offset time by t_{off}, respectively, the SAS can be described in following algorithm:

1) When a data-packet with length of x arrives at the assembler:

if ($l_b = 0$)

{$t_c = 0; l_b = x$; require scheduler to allocate outgoing channel}

else

$l_b = l_b + x$;

if ($t_c = t_{ass}$)

{report a burst is generated with length of l_b; $l_b = 0$;}

2) Scheduling outgoing channel for the generating burst:

if ($t_{ass} - t_{off} \geq 0$)

{transmission time for the burst is current time plus t_{ass}, then generate the control packet, the process is synchronous with step 1.}

else

{the transmission time for the burst is current time plus t_{off}, then generate the control packet, the process is

synchronous with step 1.}

In the above algorithm, we assume that the coming burst can get outgoing channel on required scheduling time. If it is not the case, an extra delay for the burst is need.

Pre-transmission of control packets suggests that the control packet be send during burst assembly. It is feasible because the scheduler can obtain necessary information about the coming burst for generation of the control packet before the burst is completely generated. It is originally put forward by C. Qiao[3], but it is based on estimated burst size which is very difficult for self-similar internet traffic. Here, we modify it to avoid the estimation, and make it to be the most important characteristic of the MDE scheme. To modified pre-transmission, the control packet doesn't contain the burst size, and its transmission time depends on the relationship of offset time and burst assembly time. There are two cases: one is that $t_{\rm ass}$ is greater than $t_{\rm off}$; the other is that $t_{\rm ass}$ is smaller than $t_{\rm off}$ (seen Fig. 2). The pre-transmission of control packets can be further described as follow:

1) the control packet is generated:
if $(t_{\rm ass} - t_{\rm off} > 0)$
{delay $(t_{\rm ass} - t_{\rm off})$; send the control packet;}
else
{send the control packet;}
2) the generated burst arrived:
if $(t_{\rm ass} - t_{\rm off} > 0)$
{send the burst;}
else
{delay $(t_{\rm off} - t_{\rm ass})$; send the burst;}

Generally, MDE scheme can adopt all the current OBS signaling protocol. However, the exact burst size can't be known when the control packet is generated, and the exact time when the burst assembly is finished and the offset time between control packet and burst

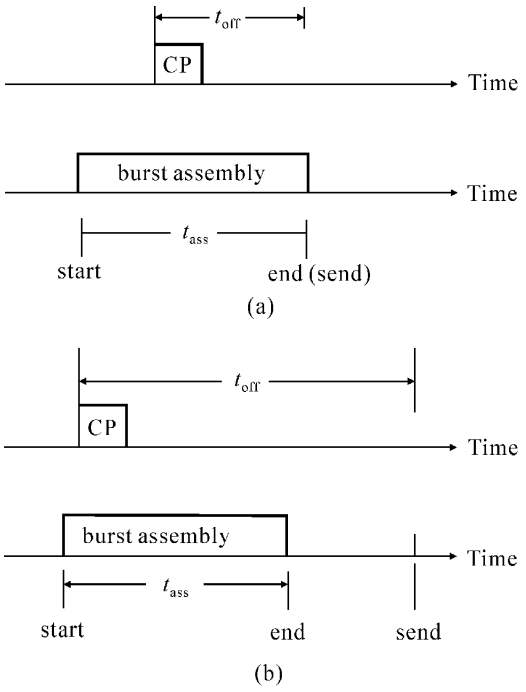

Fig. 2. Pre-transmission of control packet in two cases (CP: control packet). (a) $t_{\rm ass} > t_{\rm off}$, (b) $t_{\rm ass} < t_{\rm off}$.

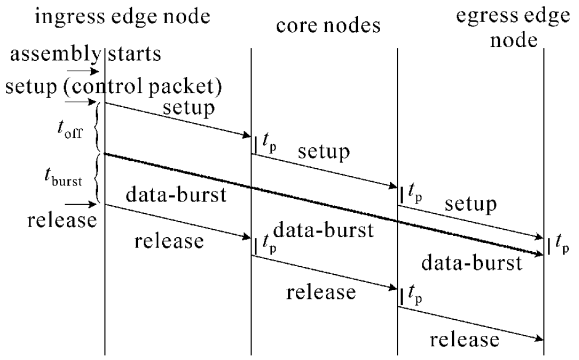

Fig. 3. The ESXR signaling protocol for the MDE scheme.

can be obtained, so the ESXR is a proper protocol for MDE scheme. In ESXR protocol, a RELEASE message is needed to release the reserved cross-connect path at core nodes after the burst has been transmitted completely[4]. Figure 3 shows the burst transmission by ESXR for MDE scheme.

To analyze performance of the MDE scheme comparing with conventional scheme, we derive the formulae for the edge delay, end-to-end latency, and the channel hold time for successful connectionless burst forwarding.

The edge delay, $t_{\rm edge}$, for an incoming data-packet is defined as the duration from the instant when the data-packet enters the ingress edge node, to the instant it starts to leave the node in a burst[5]. To a burst, the delay to its first data-packet represents the maximum to all data-packets in it. So the $t_{\rm dege}$ in this paper refers to the delay to the first data-packet in a burst.

To conventional scheme, the total delay at the edge to a data-packet can be derived as

$$t_{\rm edge} = t_{\rm ass} + t_{\rm off} + t_{\rm que} + t_{\rm proc}, \tag{1}$$

where $t_{\rm ass}$ is fixed; $t_{\rm que}$ is the duration while the burst is in the queue; $t_{\rm proc}$ is the duration for which the scheduler searches a proper outgoing channel and generates a control packet for the burst. However, utilizing MDE scheme, the total edge delay can be expressed by

$$t_{\rm edge} = \begin{cases} t_{\rm ass}, & t_{\rm ass} > t_{\rm off} \\ t_{\rm off}, & t_{\rm ass} < t_{\rm off} \end{cases}. \tag{2}$$

Thus, the edge delay is obviously shortened, comparing to the conventional scheme.

The end-to-end latency to a data-packet, $t_{\rm ete}$, is defined as the duration from the instant when the data-packet enters the ingress edge node, to the instant it is completely accepted at the egress edge node in OBS networks. It depends on the edge delay to the data-packet and its propagation in the network. We now define the notation utilized in the following analysis:

$t_{\rm etc}$: propagation delay from an edge node to its attached core node;

$t_{\rm cpp}$: protocol message processing time at each node, assuming to be identical to all nodes;

$t_{\rm cxc}$: cross-connect cut-through switching and stabilization time at a core node;

$t_{\rm ctc}$: propagation delay on a fiber link between core nodes;

t_{btx}: data-burst transmission duration.

These parameters are illustrated in Fig. 3. Assuming there are n core nodes between the ingress edge node and the egress edge node, for both the conventional scheme and MDE scheme, the end-to-end latency to a data-packet can be derived as

$$t_{\mathrm{ete}} = t_{\mathrm{edge}} + 2t_{\mathrm{etc}} + (n-1)t_{\mathrm{ctc}} + t_{\mathrm{btx}}. \quad (3)$$

Seen from this formula, MDE scheme reduces the end-to-end latency by its minimum edge delay item, comparing with the conventional scheme. Other latency items in the formula for two schemes are same.

The channel hold time at a core node, t_{cwh} , is defined to be the duration for which an outgoing channel is reserved for a data burst. It depends on the signaling protocol for OBS. For the ESXR protocol which the channel hold time at the kth core node is

$$t_{\mathrm{cwh}} = t_{\mathrm{cxc}} + t_{\mathrm{btx}} + k \cdot t_{\mathrm{cpp}} + (k-1)t_{\mathrm{ctc}} + t_{\mathrm{etc}}. \quad (4)$$

It includes the configuring time for switching matrix, the transmission time for the burst, and the process time for release message. The value is smaller than all the other signaling protocol except for JET protocol[1] which has the minimum channel hold time, $t_{\mathrm{cxc}} + t_{\mathrm{btx}}$. In fact, this is a shortcoming of the MDE scheme.

In conclusion, the MDE scheme is proposed for the OBS networks in this paper. Moreover, the performances of MDE scheme are analyzed, comparing with the conventional scheme. The results show the MDE scheme is advantaged in shortening both edge delay and end-to-end latency.

J. Yu's e-mail address is jinhui_yu@sjtu.edu.cn.

References

1. C. Qiao and M. Yoo, J. High Speed Networks **8,** 69 (1999).
2. S. Verma, H. Chaskar, and R. Ravikanth, IEEE Network **14** (Nov./Dec.), 48 (2000).
3. C. Qiao and M. Yoo, Optical Networks Magazine **1**, 36 (2000).
4. I. Baldine, G. N. Rouskas, H. G. Perros, and D. Stevenson, IEEE Communication Magazine **40** (Feb.), 82 (2002).
5. J. Y. Wei and R. I. McFarland, IEEE J. Lightwave Technol. **18,** 2019 (2000).