Hybrid approach for loss recovery mechanism in OBS networks

Ramesh P.G.V.[1*] and Prita Nair[2]

[1]*Research Scholar, Anna University and Department of Electronics and Communication,*
St. Joseph's College of Engineering, Chennai, India
[2]*Optical Research and Networks Lab, Department of Physics, SSN College of Engineering, Chennai, India*
[*]*Corresponding author: ramsdr76@gmail.com*

Received November 4, 2013; accepted February 12, 2014; posted online April 4, 2014

This letter reports a study of a hybrid burst assembly and a hybrid burst loss recovery scheme (delay-based burst assembly and hybrid loss recovery (DBAHLR)) which selectively employs proactive or reactive loss recovery techniques depending on the classification of traffic into short term and long term, respectively. Traffic prediction and segregation of optical burst switching network flows into the long term and short term are conducted based on predicted link holding times using the hidden Markov model (HMM). The hybrid burst assembly implemented in DBAHLR uses a consecutive average-based burst assembly to handle jitter reduction necessary in real-time applications, with variations in burst sizes due to the non-monotonic nature of the average delay handled by additional burst length thresholding. This dynamic hybrid approach based on HMM prediction provides overall a lower blocking probability and delay and more throughput when compared with forward segment redundancy mechanism or purely HMM prediction-based adaptive burst sizing and wavelength allocation (HMM-TP).

OCIS codes: 060.4250, 060.4510.
doi: 10.3788/COL201412.040602.

Optical burst switching (OBS), which circumvents the need for ultrafast switching and optical buffering of optical packet switching and the limitations of effective bandwidth utilization in optical circuit switching, is still the only cost-effective alternative that can cater to dynamic bandwidth on-demand applications of the present and future[1,2].

One of the major issues of the OBS network that degrades its efficiency is burst loss, when two or more bursts contend for resources at intermediate nodes. Transmission control protocol (TCP) traffic accounts for approximately 30% of real-time Internet traffic[3]. The TCP sender will be informed about the burst loss by way of a notification message transmitted through duplicate/partial ACKs as in Reno and New-Reno or by expired timers. However, in the OBS core network, its buffer-less nature and one-way just-enough-time signaling lead to arbitrary burst losses even at low traffic loads. When TCP traffic traverses over OBS networks, the random burst loss is falsely interpreted as network congestion. This false interpretation paves the way for one or more TCP false fast retransmissions (FFR), which affects the efficiency of the core network.

Efficient utilization of resources can be accomplished through burst contention mechanisms that avoid undesirable burst loss[4]. Segmentation leads to additional overheads for segments to be transmitted through the core network. Therefore, loss recovery mechanisms are needed to provide a reliable OBS transport network. The loss recovery mechanisms are broadly classified into two categories, namely, reactive and proactive. Most of the reported work that demonstrated the successful reception of the transmitted burst is mainly concerned with reactive loss recovery mechanisms. Furthermore, reactive mechanisms are invoked only after the reception of an explicit failure message, retransmits the dropped burst with additional headers for resource reservation. Reactive mechanisms are more suitable for cases where burst loss is sporadic and bandwidth utilization has to be optimized. By contrast, proactive mechanisms, which have additional payload, are appropriate when burst losses are frequent and delay is to be optimized[5]. Few loss recovery schemes for OBS networks reported use different strategies in multiple layers.

An innovative FSR mechanism presented by Chandran *et al.*[6] aims to minimize segment loss that occurs during burst contentions in the core and to recover segment loss in the forward sections from redundant segments using modified burst segmentation of each data burst.

Bikram *et al.*[7] presented a multilayer data loss recovery approach for OBS networks. They implemented an automatic retransmission request (ARQ) scheme on burst level at the lowest layer to lessen the data loss caused by random burst contentions, snoop at the next higher level to eradicate any FTOs/FFRs in the network at a packet level, and TCP retransmission of the lost packets by means of timeouts and fast retransmission mechanism at the final level.

Um *et al.*[8] proposed a priority-based duplicate burst transmission mechanism to increase the successful reception of burst by transmitting original burst and its duplicate through the same path or multipaths.

Load balancing loss recovery mechanisms are needed to overcome the lossy nature for a reliable OBS network. As discussed in existing work, loss recovery is performed proactively using redundant coding techniques or by reactively retransmitting the segments using ARQ techniques. Hence, a hybrid technique, which adaptively switches between these two techniques based on the nature of the flows, is needed.

In this letter, we proposed a hybrid burst assembly scheme with delay-based burst assembly, as suggested by Christodoulopoulos *et al.*[9], along with an additional burst length-based thresholding to restrict the variation in burst length as consecutive average delay (C_{AD}) decreases and a hybrid loss recovery mechanism. The burst assembly algorithm operates considering the C_{AD} value of the queue. While packets are entering the queue, the burst assembly algorithm measures the C_{AD} value of the queue. Once the C_{AD} reaches a predefined value $\mathrm{Th}_{\mathrm{AD}}$ or the burst length becomes equal to the predefined maximum, the burst is constructed and transmitted. The hidden Markov model (HMM) estimates the link holding time of flows. Our HMM prediction model takes link holding time of flows as hidden states and differential of the arrival and departure time of burst between node pairs in a path as observing states (P). Through a thresholding of the HMM-predicted link holding time of the optimal path assigned to a burst, the traffic that flows along this path is differentiated into long-term or short-term flows. Forward redundancy mechanism, a proactive loss recovery scheme is employed for short-term flows and a reactive scheme, which retransmits lost bursts, is used for long-term flows.

In OBS networks, for every forwarding equivalence class (FEC), a separate queue is maintained by every edge router. We let $\mathrm{Th}_{\mathrm{AD}}$ be the threshold value for the average delay of the packets in an FEC. After the deployment of nodes in the network, $\mathrm{Th}_{\mathrm{AD}}$ is defined for every FEC.

Whenever a packet enters an empty queue, the burst assembly algorithm begins the estimation of C_{AD}. At time t, the C_{AD} value can be computed as[9]

$$C_{\mathrm{AD}}(t)=\frac{D_1(t)+D_2(t)+\cdots D_{n_{\mathrm{p}}(t)}(t)}{n_{\mathrm{p}}(t)}=\frac{\sum\limits_{i=1}^{n_{\mathrm{p}}(t)}D_i(t)}{n_{\mathrm{p}}(t)}, \tag{1}$$

where $D_i(t)$ is the delay of packet i at the queue and can be obtained as

$$D_i(t)=t-A_i, \tag{2}$$

where A_i is the arrival time and $n_p(t)$ denotes the number of packets in the queue at time t.

Our burst assembly algorithm periodically estimates the value of $C_{\mathrm{AD}}(t)$. When a computed $C_{\mathrm{AD}}(t)$ reaches the $\mathrm{Th}_{\mathrm{AD}}$ value, then a burst is created, which is transmitted by the ingress node.

The burst assembly algorithm initially ends at burst accumulation time, when current C_{AD} reaches Th_{AD}. Therefore

$$C_{\mathrm{AD}}(\text{burst accumulation time})=\mathrm{Th}_{\mathrm{AD}}. \tag{3}$$

Once the burst is generated, the average delay ($C_{\mathrm{AD}}(t)$) of the queue FEC is reset to zero. This value is kept at zero until the next new packet reaches the queue.

However, this scheme has the drawback of creating large variations in burst sizes. Considering C_{AD} as a function of time, C_{AD} can be calculated at time $t+\gamma t$ as

$$C_{\mathrm{AD}}(t+\gamma t)=\frac{n_{\mathrm{p}}(t)\cdot(C_{\mathrm{AD}}(t)+\gamma t)}{n_{\mathrm{p}}(t+\gamma t)}. \tag{4}$$

If no packet arrives in this FEC at the time interval γt, then the value of C_{AD} becomes

$$C_{\mathrm{AD}}(t+\gamma t)=C_{\mathrm{AD}}(t)+\gamma t. \tag{5}$$

In this case, $C_{\mathrm{AD}}(t)$ increases in proportion to time with the slope of one and gradually reaches the threshold value $\mathrm{Th}_{\mathrm{AD}}$.

By contrast, when multiple packets reach the queue at the interval γt, the average delay will be smaller and, hence, the C_{AD} will take a longer time to reach the Th_{AD} value, i.e., $C_{\mathrm{AD}}(t)$ increases at a slower rate. Hence, to restrict the burst accumulation time in this scenario, burst assembly in our algorithm is modified to occur if either $C_{\mathrm{AD}}=\mathrm{Th}_{\mathrm{AD}}$ or Burst length=Maximum burst length, whichever happens first, as described in Algorithm 1 shown in Fig. 1.

The holding time of a link is the resource utilization time of the link. Future connection arrival instants and holding times may not be known in advance. The link holding time ($L_{\mathrm{h}}t$) can be estimated from the burst arrival time ($B_{\mathrm{a}}t$) at node N_i and burst departure time ($B_{\mathrm{d}}t$) at node (N_{i-1}) as

$$L_{\mathrm{h}}t=B_{\mathrm{d}}t-B_{\mathrm{a}}t. \tag{6}$$

We let $B_{\mathrm{a}}tk(j)$ and $B_{\mathrm{d}}tk(j)$ be the burst arrival and burst departure times, respectively, of burst j at link Li at time tk, where $i=1,2,\cdots,n$ (hops), then the holding time of link Li by burst j can be expressed as

$$L_{\mathrm{h}}tk\,(Li)\,(j)=B_{\mathrm{d}}tk\,(j)-B_{\mathrm{a}}tk\,(j)\,. \tag{7}$$

Then, the total link holding time of burst j at time $tk+1$, $tk+2$ can be predicted using the HMM.

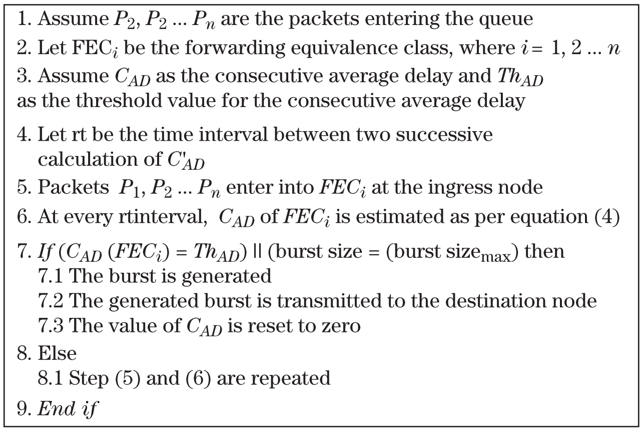
Our HMM prediction model takes link holding time of future traffic of flows as hidden states and burst arrival and burst departure times of current traffic as observing states (P).

We let H be the set of hidden states expressed as

$$H=h_1,h_2,\cdots,h_n. \tag{8}$$

We let P be the set of observation states expressed as

$$P=p_1,p_2,\cdots,p_n. \tag{9}$$



```
1. Assume $P_2, P_2 \ldots P_n$ are the packets entering the queue
2. Let FEC$_i$ be the forwarding equivalence class, where $i = 1, 2 \ldots n$
3. Assume $C_{AD}$ as the consecutive average delay and $Th_{AD}$
as the threshold value for the consecutive average delay
4. Let rt be the time interval between two successive
   calculation of $C'_{AD}$
5. Packets $P_1, P_2 \ldots P_n$ enter into $FEC_i$ at the ingress node
6. At every rtinterval, $C_{AD}$ of $FEC_i$ is estimated as per equation (4)
7. If ($C_{AD}$ ($FEC_i$) = $Th_{AD}$) || (burst size = (burst size$_{\mathrm{max}}$) then
   7.1 The burst is generated
   7.2 The generated burst is transmitted to the destination node
   7.3 The value of $C_{AD}$ is reset to zero
8. Else
   8.1 Step (5) and (6) are repeated
9. End if
```

Fig. 1. Burst assembly algorithm.

We let C be the state sequence of length L corresponding to observation P expressed as

$$C = c_1, c_2, \cdots, c_L. \tag{10}$$

During the prediction interval time Tp, each intermediate node along the path of the given source and destination pair estimates the link holding time and passes that value to the HMM prediction model as an observation state. HMM predicts the link holding time (H) of the flows for the future interval with the observation sequence C. Finally, from the destination, the connection holding time of the entire path is forwarded to the source node.

The probability of the observation (P) in a given sequence C is expressed as[10]

$$\begin{aligned} Pr\,(P|C,\lambda) &= \prod_{l=1}^{L} Pr\,(p_l|c_l,\lambda) \\ &= b_{c1}(p_1) \times b_{c2}(p_2), \cdots, b_{cL}(p_L). \end{aligned} \tag{11}$$

The state sequence probability is expressed as

$$Pr(C|\lambda) = \pi_{c1} a_{c1c2} a_{c2c3} \ldots a_{cL-1cL}. \tag{12}$$

We can easily estimate the probability of observations using

$$\begin{aligned} P(P|\lambda) &= \sum_{N} Pr\,(P|C,\lambda) Pr\,(C|\lambda) \\ &= \sum_{c1\ldots cL} \pi_{c1} b_{c1}(p_1) a_{c1c2} b_{c2}(p_2) \ldots a_{cL-1cL} b_{cL}(p_L). \end{aligned} \tag{13}$$

HMM uses the Viterbi algorithm to determine the solitary state sequence for an observation sequence c_1. To determine the higher likelihood state, we first outline the probability of the most possible path as[10]

$$\varsigma(i) = \max_{c1,c2,c3,\ldots nL-1} P\left(c_1, c_2 \cdots c_L = P_i, p_1, p_2 \ldots p_l|\lambda\right). \tag{14}$$

By using the aforementioned probability function, we can determine the higher likelihood state as

$$\overset{*}{n}_L = \arg \max_{1 \leqslant i \leqslant L} [\varsigma_L(i)]. \tag{15}$$

At each step, the sequence of states can be backtracked as the pointer. The backtracking process of the state sequence is expressed as[10]

$$\overset{*}{n}_L = \Psi_{l+1}(\overset{*}{n}_{l+1}), \quad l = L-1, L-2, \cdots, 1, \tag{16}$$

where Ψ is an additional matrix of $C * L$; this matrix should be added in the Viterbi algorithm to the optimal state. L denotes the state sequence length time. This backtracking provides the required set of states.

The paths are classified into long-term and short-term paths by comparing computed connection holding (l_{h}) time with predefined threshold value TL_{h}. Flows that have connection holding time lesser than or equal to TL_{h} are marked as short-term flows. By contrast, flows that have connection holding time greater than TL_{h} are termed as long-term flows. This classification into long-term and short-term traffic based on link holding times predicted using HMM is further utilized to implement our hybrid loss recovery scheme which uses a reactive scheme for long-term traffic and proactive scheme for short-term traffic.

The proactive scheme, which improves the performance of short-term bursty traffic flows of the OBS network by reducing packet loss, uses a forward redundancy mechanism. While transmitting burst between the source and the destination, the forward redundancy scheme duplicates (copies) a few or all packets of the burst and sends them in the forward direction along with the original burst. In the proposed technique, the redundant packets are transmitted to the destination following a serial forward redundancy scheme, (i.e., the replicated data is appended at the tail end of the original burst in a serial manner) such that destination can recover the packets lost from the original burst. Typically, the two kinds of redundancy schemes are as follows: complete forward redundancy, in which the redundant segment is equal to 100% of the original burst, and partial forward redundancy, in which the redundant segment is <100% of the original burst. The selection between forward and partial redundancy schemes can be made considering the requirements of data traffic. In our technique, complete forward redundancy is implemented.

To enhance the performance of long-term flows on the OBS network, the reactive loss recovery mechanism is used. Before transmitting data burst, the source forwards the burst header packet (BHP) in the network to reserve resources along the path and the burst is forwarded after the expiration of an offset timer. While transmitting data, the source keeps track of a copy of the transmitting burst so that it can enable the retransmission of burst upon failure. On traversing through the core nodes, if the BHP discovers the channel reservation failure caused by a burst contention, then it immediately forwards back an ARQ to the source. By receiving an ARQ message, the source retransmits the failed burst, which is led by a duplicate BHP.

The flow classification and loss recovery algorithm of the proposed mechanism is described in the flowchart shown in Fig. 2.

The performance of delay-based burst assembly and hybrid loss recovery (DBAHLR) mechanisms is examined for the mesh topology (Fig. 3) using the Network Simulator-2 (NS-2) simulator[11] with ORIC Obs-0.9a extension.

The simulation settings utilized in this analysis are as follows: the total number of edge and core nodes is 14, respectively, and the maximum number of channels is 10, with 2 for control wavelengths and 8 for data channels. Channel bandwidth is 100 Mbps, and the traffic load is expressed as packet sending rate measured in Mbps per ingress node. Simulations were conducted by varying the traffic load from 14 to 28 Mbps per ingress node.

In this simulation, a self-similar traffic model is used for short-term traffic and TCP is used for long-term traffic. Five TCP traffic flows are set up between pairs of ingress and egress edge routers with four sets of simultaneous short-term traffic flows. In all simulations, the results of DBAHLR are compared with the FSR and HMM prediction-based adaptive wavelength allocation and burst sizing without loss recovery scheme called

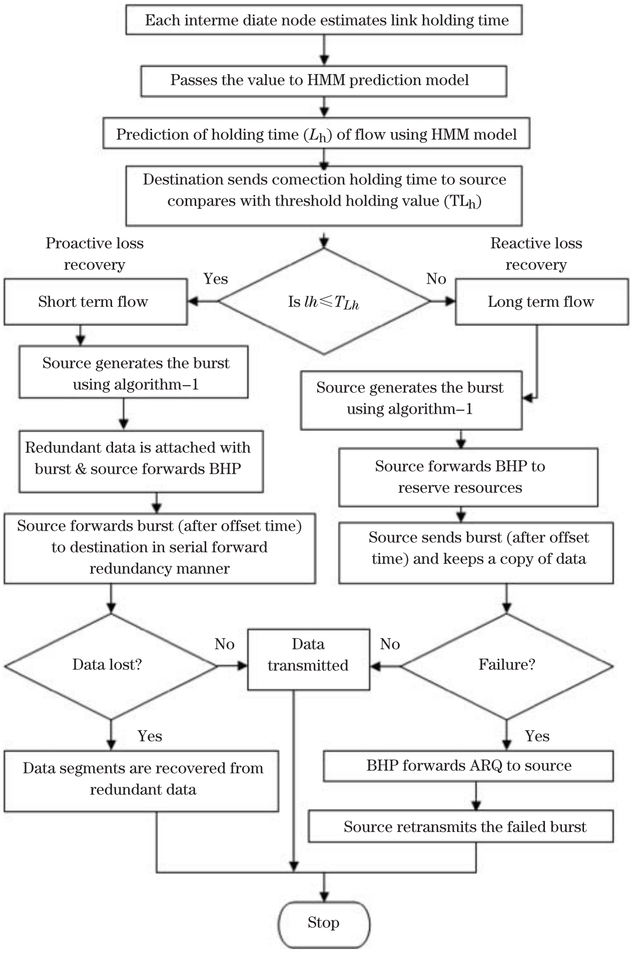

Each interme diate node estimates link holding time
Passes the value to HMM prediction model
Prediction of holding time (L_h) of flow using HMM model
Destination sends comection holding time to source compares with threshold value (TL_h)
Proactive loss recovery
Reactive loss recovery
Yes
No
Is $lh\leqslant T_{Lh}$
Short term flow
Long term flow
Source generates the burst using algorithm−1
Source generates the burst using algorithm−1
Redundant data is attached with burst & source forwards BHP
Source forwards BHP to reserve resources
Source forwards burst (after offset time) to destination in serial forward redundancy manner
Source sends burst (after offset time) and keeps a copy of data
Data lost?
No
Data transmitted
No
Failure?
Yes
Yes
Data segments are recovered from redundant data
BHP forwards ARQ to source
Source retransmits the failed burst
Stop

Fig. 2. Flow classification and loss recovery technique.

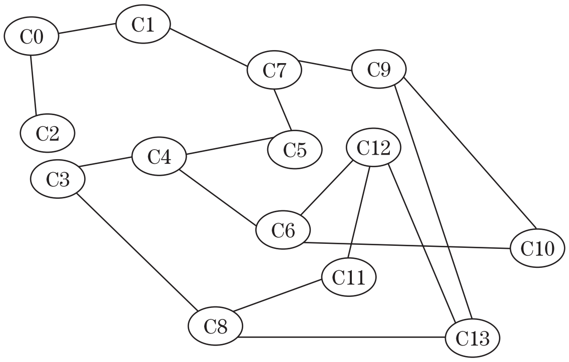


Fig. 3. Network topology.

the HMM-TP[12] method.

Figure 4 shows the blocking probability of the DBAHLR, FSR, and HMM-TP techniques for different load scenarios. As the traffic load increases, the blocking probability increases because of congestion and overloading. The blocking probability of the proposed DBAHLR approach is 23% less than the FSR approach and only marginally higher than the HMM-TP approach. The reduction in blocking probability of DBAHLR and HMM-TP over FSR is attributed to the efficiency of the underlying HMM-predicted traffic classification and associated provisioning. The marginal increase in blocking probability of DBAHLR over HMM-TP is due to the additional payloads and BHPs because of the loss recovery schemes.

Figure 5 shows the burst delay for the DBAHLR, FSR, and HMM-TP techniques in different load scenarios. The burst delay of the proposed DBAHLR approach is 19% less than the FSR approach and approximately 22% less than the HMM-TP approach. This reduction in delay of DBAHLR is clearly due to the hybrid burst assembly technique. Figure 6 shows the number of bursts received for the DBAHLR, FSR, and HMM-TP techniques. This study shows that the burst received in the proposed DBAHLR approach is 30% higher than the FSR approach and approximately 40% higher than the HMM-TP approach. The higher throughput of DBAHLR when compared with HMM-TP is probably due to the combined effect of reduction in contentions because of lesser delays and also the hybrid loss recovery techniques.

The performance of short-term and long-term traffic has been segregated to assess the effect of the DBAHLR scheme on each of them independently. Figure 7 shows the throughput of the DBAHLR and FSR techniques

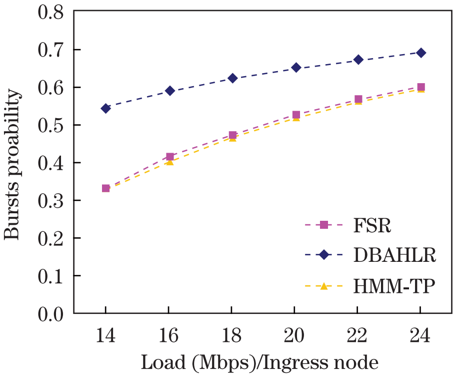


Fig. 4. Blocking probability.

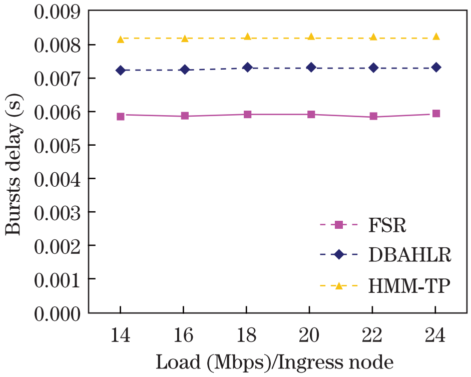


Fig. 5. Burst end-to-end delay.

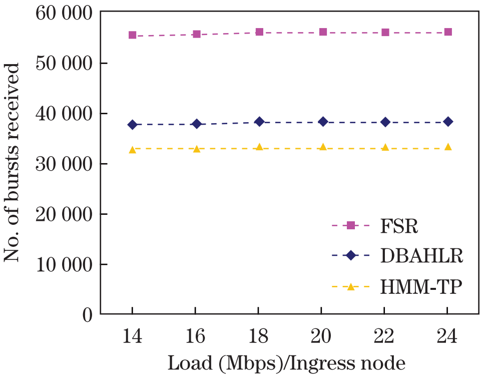


Fig. 6. Burst received for all traffic flows.

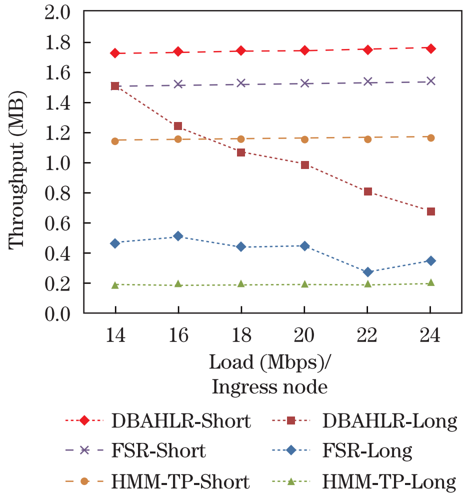


Fig. 7. Throughput for short-term and long-term flows.

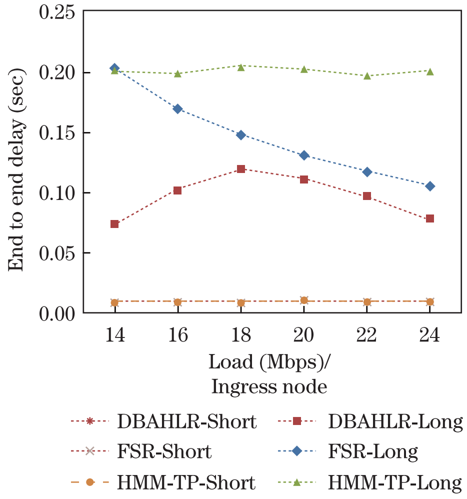


Fig. 8. End-to-end delay for short-term and long-term flows.

separately for the long-term and short-term traffic for different load scenarios. The long-term traffic throughput of our proposed DBAHLR approach is 60% higher than the FSR approach and 80% higher than the HMM-TP approach. The short-term traffic throughput of our proposed DBAHLR approach is 12% higher than the FSR approach and approximately 33% higher than the HMM-TP approach. This result indicates that loss recovery has effectively increased the throughput.

Figure 8 shows the end-to-end delay of the DBAHLR, FSR, and HMM-TP techniques for the short-term and long-term traffic flows at different load scenarios. The short-term traffic delay of the proposed DBAHLR approach is only marginally lesser than the FSR approach, but is less by 6% when compared with the HMM-TP approach. The long-term traffic delay of our proposed DBAHLR approach is 30% lesser than the FSR approach and 60% lesser than the HMM-TP approach. The reduction in delay with increasing load observed for long-term traffic may be due to faster burst assembly and restricted burst length, which, in turn, reduces burst loss and, hence, retransmission attempts. This study confirms that the hybrid burst assembly scheme along with hybrid loss recovery reduces the delay of bursts and improves the throughput. In addition, the effect on long-term traffic is much higher when compared with that on short-term traffic.

In conclusion, we evaluated the performance of the hybrid average delay-cum-burst size based burst assembly and hybrid loss recovery mechanisms for OBS networks. Our burst assembly algorithm operates considering the C_{AD} value of the queue. While the packets are arriving into the queue, the burst is constructed if the estimated C_{AD} reaches the threshold value or the burst size approaches a maximum value, restricting burst size variation inherent in C_{AD} thresholding. The HMM prediction model predicts the link holding time available in each path and classifies the possible flows in these paths into long-term and short-term flows with associated resource allocation strategies. A proactive loss recovery scheme, that is, the forward redundancy mechanism, is used for short-term flows. A reactive scheme, which retransmits lost bursts, is used for long-term flows. The proposed mechanisms are simulated in NS-2. An overall improvement in delay of 19% and a throughput improvement of 30% over FSR have been obtained. Further analysis reveals that the effect of the hybrid burst assembly and loss recovery scheme is actually more for the long-term traffic flows. Further reduction in the delays for long-term traffic, which in our case is approximately 30 times higher than that of short-term traffic, can be achieved if bandwidth-variable OBS schemes can be implemented selectively for long-term flows in this classification scheme.

References

1. Y. Chen, C. Qiao, and X. Yu, IEEE Network **18**, 16 (2004).
2. X. Cao, B. Wu, X. Hong, and J. Wu, Chin. Opt. Lett. **10**, 070606 (2012).
3. W. John, M. Dusi, and K. C. Claffy, in *Proceedings of the 6th International Wireless Communications and Mobile Computing ACM Conference* 473 (2010).
4. A. K. Garg and R. S. Kaler, Chin. Opt. Lett. **6**, 807 (2008).
5. V. M. Vokkarane and Q. Zhang, in *Proceedings of IFIP International Conference on Wireless and Optical Communications Networks* (2006).
6. D. Chandran, N. Charbonneau, and V. M. Vokkarane, in *Proceedings of IEEE 2nd International Symposium on Advanced Networks and Telecommunication Systems, ANTS '08* 1 (2008).
7. R. R. C. Bikram, N. Charbonneau, and V. M. Vokkarane, J. Photon. Network Commun. **21**, 158 (2011).
8. T. W. Um, H. L. Vu, J. K. Choi, and W. Ryu, ETRI Journal 30, 164 (2008).
9. K. Christodoulopoulos, E. Varvarigos, and K. Vlachos, Elsevier, Optical Switching and Networking **4**, 200 (2007).
10. P. Blunsom, "Hidden Markov Models" The University of Melbourne, Department of Computer Science, www.digital.cs.usu.edu (2004).
11. Network Simulator: http:///www.isi.edu/nsnam/ns
12. P. G. V. Ramesh and P. Nair, J. Comput. Sci. **10**, 821 (2014).