

Bandwidth-Efficient Dynamic Tree-Shared Multicast in Optical Burst-Switched Networks*

Myoungki Jeong and Chunming Qiao
Depts. of EE and CSE, SUNY at Buffalo
Buffalo, NY 14260
{mjeong,qiao}@cse.buffalo.edu

Yijun Xiong
Mahi Networks
Petaluma, California 94954
yxiong@mahinetworks.com

Marc Vandenhoute
Alcatel Network Strategy Group
Richardson, TX 75081
vandma@usa.alcatel.com

Abstract— In this paper, we study three multicast schemes, namely Separate Multicasting (S-MCAST), Multiple Unicasting (M-UCAST), and Tree-Shared Multicasting (TS-MCAST), in optical burst-switched WDM networks taking into consideration the overheads due to control packets and guard bands (GBs) of bursts on separate channels (wavelengths). In TS-MCAST, we describe four tree sharing strategies based on Equal Coverage (EC), Super Coverage (SC), Overlapping Coverage (OC) and Overlapping Coverage by Maximization (OC-MAX) for deciding which multicast sessions should mix their multicast traffic, and also consider an algorithm to construct shared trees (STs). In [1], the tree sharing strategies are proposed and the performance of three multicast schemes for static multicast sessions and membership is reported. In this paper, we propose efficient heuristic algorithms for managing dynamic sessions and memberships under the TS-MCAST scheme, and evaluate the efficiency of the proposed heuristic algorithms and compare the TS-MCAST scheme with the other two schemes in terms of bandwidth consumed and processing load assuming the unlimited bandwidth.

1 Introduction

As traffic demand increases exponentially in the Internet, Wavelength Division Multiplexing (WDM) networks [3, 4, 5] become a natural choice for the backbone. Recently, IP over WDM networks (or so-called Optical Internet) have received a considerable amount of attention (e.g. [6, 7]).

Multicasting (i.e. one-to-many or many-to-many communications) is becoming more and more popular and important in the Internet. Multicasting in IP over WDM networks can be done via IP multicast, multiple WDM unicast, or WDM multicast [8]. In this paper, we will concentrate mainly on WDM multicast.

There are two WDM multicasting approaches, one based on wavelength-routing as in [9, 10, 11], and the other based on optical burst switching (OBS) [7, 12] as in [8, 13]. In the former, a multicast tree is built for each multicast session by reserving one wavelength on each of its branches for the duration of the session. In the latter, no wavelengths need to be dedicated to a mul-

ticast tree and the multicast data is transported via OBS which is more bandwidth efficient than wavelength routing for bursty traffic. However, there are two major overheads in using OBS, namely, control packets and guard bands. More specifically, to send each burst (for unicast or multicast), a control packet needs to be sent to set up switches and guard bands are used in the burst to accommodate possible timing jitters at each intermediate node. Issues related to these overheads have not been considered in [8, 13].

In this paper, we study several multicast schemes in optical burst-switched WDM networks taking into consideration the overheads due to control packets and guard bands (GBs) of bursts on separate channels (wavelengths). These will result in different bandwidth consumption (which is proportional to different amount of GBs), different channel utilization (inversely proportional to different burst lengths) and different processing loads (proportional to the number of control packets generated) under the same traffic condition. Specifically, we consider three multicast schemes. A straightforward scheme is called Separate Multicasting (S-MCAST) where each source node constructs separate bursts for its multicast and unicast traffic. In other words, a burst will contain either multicast traffic only or unicast traffic only. To reduce the overhead due to GBs (and control packets), one may piggyback the multicast traffic in bursts containing unicast traffic using a scheme called Multiple Unicasting (M-UCAST). In other words, multicast is accomplished with multiple unicast and hence multicast traffic essentially gets a “free” ride. However, the overall bandwidth utilization may be low due to multiple unicast transmissions of the same multicast data. The third scheme is called Tree-Shared Multicasting (TS-MCAST) whereby multicast traffic belonging to multiple multicast sessions can be mixed together in a burst, which is delivered via a *shared* multicast tree. In TS-MCAST, we describe four tree sharing strategies based on Equal Coverage (EC), Super Coverage (SC), Overlapping Coverage (OC) and Overlapping Coverage by Maximization (OC-MAX) for deciding which multicast sessions should mix their multicast traffic, and also describe an algorithm to construct shared trees (STs).

In [1], the tree sharing strategies are proposed and the performance of three multicast schemes for static multicast sessions and membership is reported. In this paper, we focus on the cases with dynamic birth and death of a session, and dynamic join/leave of a member in existing multicast sessions. We pro-

*This research is supported in part by a grant from NSF under contract number ANIR-9801778.

pose efficient heuristic algorithms for managing dynamic sessions and memberships under the TS-MCAST scheme. The performance of three multicast schemes with limited bandwidth and dynamic sessions using heuristic algorithms proposed here is reported in [2] focusing on the burst blocking probability. In this paper, we evaluate the efficiency of the proposed heuristic algorithms and compare the TS-MCAST scheme with the other two schemes in terms of bandwidth consumed and processing load assuming the unlimited bandwidth.

The rest of this paper is organized as follows. In Section 2, we describe tree sharing strategies and an algorithm to construct shared trees. In Section 3, we propose several heuristic algorithms to support dynamic sessions and membership for the TS-MCAST scheme. In Section 4, we describe the simulation model used and present performance of multicast schemes using the proposed heuristic algorithms under dynamic sessions/membership. In Section 5, we conclude the paper.

2 Tree-Shared Multicasting in OBS Networks

In this Section, we describe tree sharing strategies whereby a set of multicast sessions (H_i) at edge router i is decomposed into a number of Multicast Sharing Classes (MSCs)¹ where each MSC uses a ST, and a shared tree construction algorithm in the TS-MCAST scheme.

Let N_c be the set of all core routers, N_e be the set of all edge routers and N_l be the set of all links in the network. In the following discussion, we model a multicast session (or tree) in the network using a triple $\mathbf{T} = (C, E, L)$ where $C \subseteq N_c$ is the set of core routers, $E \subseteq N_e$ is the set of edge routers, and $L \subseteq N_l$ is the set of links on the multicast tree, respectively.

2.1 Tree Sharing Strategies

We consider four strategies, namely Equal Coverage, Super Coverage, Overlapping Coverage and Overlapping Coverage by Maximization for deciding which subset of $|H_i| (\geq 2)$ multicast sessions rooted at edge router i should become a MSC. For simplicity, we use MSC_j rather than MSC_{ij} to denote the j th MSC at edge router i hereafter.

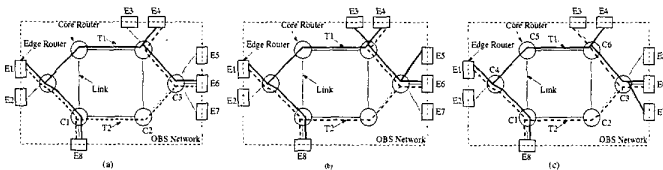


Figure 1: Tree Sharing Strategies: (a) Equal Coverage, (b) Super Coverage, and (c) Overlapping Coverage.

¹subset and MSC will be used interchangeably and as a special case, a MSC may contain only one multicast session.

In EC, multicast sessions with the same membership are grouped into one MSC. In other words, the $s \geq 2$ multicast sessions (T_1 through T_s) in MSC_j have the same set of member edge routers, i.e., $E_{T_1} = E_{T_2} = \dots = E_{T_s}$, although each multicast session in MSC_j may have a different multicast tree (or path to each member). Fig. 1 (a) shows an example of EC (where $s = 2$) in which multicast trees T_1 (solid line) and T_2 (dashed line) have the same set of edge routers (i.e., E4, E6 and E8) as their members. In such a case, one of the existing multicast trees, T_1 or T_2 , is selected to be the new ST.

A less restricted tree sharing strategy is SC where the multicast sessions in MSC_j do not necessarily have the same set of edge routers. Specifically, if two multicast sessions, T_1 and T_2 , are such that $E_{T_2} \subseteq E_{T_1}$ (or $E_{T_1} \subseteq E_{T_2}$), then T_1 and T_2 are grouped into one MSC (see Fig. 1 (b)). Note that in Fig. 1 (b), IP packets belonging to T_2 will also be delivered to E3 and E5 via T_1 since $T_2 \subseteq T_1$, but subsequently discarded by E3 and E5.

The third scheme OC is a more general tree sharing strategy in which it allows a number of multicast sessions having a sufficient degree of overlap in the set of edge routers E_T , core routers C_T , or links L_T or a subset of multicast sessions resulting in a sufficient tree sharing gain α (from (2)) to be grouped into one MSC. More specifically, we define the degree of overlap as follows. Consider s multicast trees (sessions), $\mathbf{T}_k = (C_{T_k}, E_{T_k}, L_{T_k})$ for $k = 1, 2, \dots, s$. Then, the degree of overlap among these s multicast sessions can be defined in terms of edge routers as follows:

$$\gamma^E = \frac{\sum_{k=1}^s |E_{T_k}| - |\bigcup_k E_{T_k}|}{|\bigcup_k E_{T_k}| (s-1)}. \quad (1)$$

Note that according to such a definition, $\gamma^E = 1$ in EC and $\gamma^E \leq 1$ in SC and OC. Similarly, the degree of overlap is defined in terms of core routers and links and they are given in [1].

One may apply the OC strategy as follows. Let $R \subseteq H_i$ be a set of remaining multicast sessions to be evaluated and initially $R = H_i$. First, select one of the above three criteria, namely, the edge router overlap degree (γ^E), core router overlap degree (γ^C) or link overlap degree (γ^L). Then from the set R , a pair of sessions with the highest value, say γ_1 , of γ^E , γ^C , or γ^L (depending on the criterion selected) is chosen to form a temporary MSC (say MSC_1), which also becomes a candidate MSC_{cand} . Afterwards, a third session that results in the largest value, say γ_2 , in the degree of overlap when combined with the pair (forming a temporary MSC_2) is selected. If $\gamma_2 \geq \gamma_1$, the candidate MSC is replaced with MSC_2 (i.e., now it contains three multicast sessions). Otherwise, the candidate MSC is still MSC_1 . The same process is repeated for the fourth session in the set (if any, which does not belong to MSC_2) with MSC_2 , and so on. Finally, the formed candidate MSC (i.e., MSC_{cand} , its size ≥ 2) becomes a new MSC only if the tree sharing gain α for MSC_{cand} is above the threshold η . Note that when α for MSC_{cand} is below the threshold η , MSC_{cand} is put into a set U which contains unshared multicast sessions. The same procedure above is repeated for remaining multicast sessions $R (= R - MSC_{cand})$ to form another MSC until $|R| = 1$. Finally, each unshared multicast session in U and R becomes a MSC with size 1. An example where the OC strategy

can be used is shown in Fig. 1 (c) as T_1 and T_2 have member edge routers that do not belong to the other sessions. The OC strategy requires $O(M^2)$ criterion checks where $M(=|H_i|)$ is the number of multicast sessions at edge router i .

Note that in both SC and OC, some data (IP packets within a burst) may be delivered to unintended destination (egress) edge routers where these IP packets will simply be discarded.

The OC sharing strategy described above does not guarantee the optimal selection of subsets (MSCs) so that the sharing gain at the edge router is maximized. Instead, it tries to select the best MSC that has maximum value of a criterion applied. Here we consider a new OC sharing strategy, called OC-MAX, to maximize the sharing gain at the edge router. In OC-MAX, it searches for a combination of disjoint subsets (MSCs) from all possible subsets so that the combination can provide the maximum sharing gain achievable at the edge router based on an exhaustive search.

For a MSC j at an edge router i , denoted by MSC_{ij} , we define the bandwidth gain due to tree sharing as the ratio between the average amount of multicast traffic carried per link with/without tree sharing. Let r_{ij} be the data rate of a session j at an edge router i and s_{ij} be the number of links on the multicast tree for the session j . In addition, let r'_{ij} be the sum of the data rate of all the sessions in MSC_{ij} and s'_{ij} be the number of links on the shared multicast tree used for MSC_{ij} . Assuming that G is the guard band size, then the bandwidth gain α_j for MSC_{ij} due to tree sharing is equal to

$$\alpha_{ij} = \frac{\sum_{k \in MSC_{ij}} (r_{ik} + G/T_b^m) s_{ik}}{(r'_{ij} + G/T_b^m) s'_{ij}}. \quad (2)$$

2.2 Construction of Shared Trees

After one has decomposed a set of multicast sessions (H_i) at edge router i into a number of MSCs according to one of the tree sharing strategies, each MSC has to construct a ST by treating all the members in the subset of the multicast sessions as a new multicast group for the purpose of burstification (forming bursts) and burst delivery. Note that the cases for EC or SC are trivial because one can use any existing tree in EC, and a super tree in SC, respectively. For OC (and OC-MAX), we proposed ST construction algorithms [1] to construct a ST $T^S = (V^S, L^S)$ where V^S is the set of all nodes (edge and core routers), and L^S is the set of all links on the ST, respectively. Among them, the ST-MEMBER applies to OC and OC-MAX strategies to construct the shared tree of a MSC in this paper.

3 Dynamic Sessions and Membership

3.1 Dynamic Sessions

To support the dynamic birth/death of a session with fixed membership in tree-shared multicasting, we propose the following heuristic algorithms for the OC sharing strategy where a session is added (or deleted) dynamically to (or from) existing multicast sessions at the edge router. A similar heuristic algorithm can be

applied for the EC/SC sharing strategy except that gain check is not performed, instead, the criterion for EC/SC is examined. It is assumed that a set of multicast sessions, H_i at the edge router i is decomposed into a set of subsets (size of each ≥ 2), S and a set of subsets (size of each = 1), U .

A. Birth Operation

If a new session n is created with pre-determined membership, the new session n is evaluated with existing MSCs (size of each ≥ 2). If the new session can fit in one of them, then add n to the selected session. A detailed description of the heuristic algorithm is as follows.

SESSION-ADD Algorithm

S = set of MSCs (size of each ≥ 2); U = set of MSCs (size of each = 1);

$H_i = S \cup U$;

a new session n is created;

for each $MSC_j \in S$,

add the new session n to $MSC_j \in S$, i.e., $MSC'_j = MSC_j \cup n$;

Compute_Gain(MSC'_j); // gain α'_j

if ($\alpha'_j > \eta$)

add n to MSC_j , i.e., $MSC'_j = MSC_j \cup n$;

for each session $u \in U$

evaluate u with MSC'_j , i.e., $MSC''_j = MSC'_j \cup u$;

Compute_Gain(MSC''_j); // gain α''_j

if ($\alpha''_j > \eta$)

add u to MSC'_j ;

$U \leftarrow U - \{u\}$;

end if

end for

end if

end for

if no $MSC_j \in S$ to add the new session n ,

put n into U , i.e., $U' \leftarrow U \cup \{n\}$;

apply tree sharing strategy to U' by choosing one of criteria;

end if

B. Death Operation

If a session in U expires, just delete it and its related state information. However, if a session $m \in MSC_k$ (≥ 2) ends, first MSC'_k is checked to see if it can still provide a sharing gain ($> \eta$). If so, keep it as a new MSC. Otherwise, MSC'_k is evaluated with existing MSCs. A detailed pseudo code describing a heuristic algorithm called SESSION-DELETE algorithm is given in Appendix A.

As described above, note that if a newly created multicast session is added to one of existing MSCs (say MSC_j), it is needed to evaluate each session in U to see if it can fit in MSC_j . In addition, if a session expires in a MSC (say MSC_k) and some of the remaining multicast sessions in MSC_k after forming new MSCs join existing MSCs (making newly formed MSCs), it is also needed to evaluate each session in U with newly formed MSCs. To illustrate these points, suppose that there is a set $H_i = \{G_1, G_2, G_3, G_4, G_5, G_6, G_7, G_8, G_9, G_{10}\}$ at an edge router i . Assume that by applying the OC sharing strategy the set H_i is split as follows: $MSC_1 = \{G_3, G_4\}$, $MSC_2 = \{G_5, G_6, G_1, G_{10}\}$, $MSC_3 = \{G_7, G_8\}$, $MSC_4 = \{G_2\}$ and $MSC_5 = \{G_9\}$ ($U = MSC_4 \cup MSC_5$). First, if a session G_{11} is newly created and it fits in MSC_1 , $MSC'_1 (= MSC_1 \cup G_{11})$ needs to be evaluated with each session in U

because G_{11} is not evaluated with a session in U before. On the other hand, if a session G_1 in MSC_2 expires, the remaining sessions in MSC_2 are G_5, G_6 and G_{10} . Suppose that (G_1, G_{10}) was the best pair of $H_i - MSC_1$, and then G_5/G_6 were added to the best pair (G_1, G_{10}) as the third and fourth session. If G_5 and G_6 form a new MSC and the session G_{10} is added to MSC_3 , MSC_3 contains G_7, G_8 and G_{10} . MSC_3 is a new subset which is not evaluated previously according to the OC sharing strategy. In fact, (G_7, G_8) is compared with the best pair (G_1, G_{10}) and (G_7, G_8, G_{10}) is not evaluated since the value of applied criterion for the degree of overlap is smaller than that of the pair (G_1, G_{10}) , thus (G_7, G_8) is not selected. Therefore, each session in U needs to be evaluated with MSC_3 . In addition, the pair (G_5, G_6) is not evaluated with each session in U previously, hence it needs to be evaluated.

3.2 Dynamic Membership

In IP multicasting, a member (edge router) in a session may leave the session, or a new member may join the session at any time. To support dynamic join/leave in tree-shared multicasting, we propose two approaches: time-based approach and need-based approach. In the former, all MSCs are evaluated at regular intervals to see if they can still provide any sharing gain. In the latter, All MSCs are evaluated only after there is a significant change in membership. First, we differentiate the dynamic behavior into two cases.

A. Incremental Change

If a new edge router, E_j , joins one of sessions (say g) in MSC_i , which already has joined another session(s) (other than the session g) of MSC_i , the degree of overlap for MSC_i will increase (sharing gain accordingly). In addition, if an edge router E_j leaves one of sessions (say g), in the MSC_i and g is the only session in the MSC_i that has the edge router E_j as one of its members, then the degree of overlap for the MSC_i also increases. We call this type of join/leave *incremental change* in membership. As an example, suppose that MSC_i has three multicast sessions (see Figure 2). If an edge router, E_4 (or E_1), in Figure 2 joins (or leaves) the multicast group G_3 , the degree of overlap will increase. If the incremental change happens in a MSC, the MSC does not change.

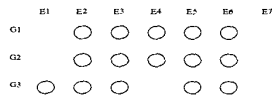


Figure 2: A membership example of a Multicast Sharing Class.

B. Decremental Change

If dynamic join/leave decreases the degree of overlap for a MSC (say MSC_i), we call this type of join/leave *decremental change* in membership. As an example, if an edge router E_7 (or E_3) joins (or leaves) a multicast group G_3 in Figure 2, the degree of overlap for the MSC will decrease. A MSC that has

the decremental change needs to be evaluated based on certain criteria so that it may be groomed differently (to still provide the sharing gain larger than η).

Note that the same argument for incremental/decremental change can apply to the degree of overlap using link/core. As an example, let us consider a simple network in Figure 3. Suppose MSC_i has two multicast sessions, T_1 and T_2 . If a new member (edge router E_7) joins the multicast tree T_1 , the link L_1 (core C_2) will be included to T_1 . This is the case for incremental change where the degree of overlap using link (or core) increases, thus increasing the sharing gain α . In addition, if a member (E_7) leaves the multicast tree T_2 , this is also the case for incremental change in link (core). On the other hand, if a new member, edge router E_8 , joins the multicast tree T_2 , a new branch (shown in dotted line) will be added to T_2 . This is the case for decremental change which results in adding a new link L_2 (a new core router C_3) to T_2 , thus decreasing the degree of overlap using link/core. Note that if there is an increase (decrease) in the degree of overlap (using member, link or core) in MSC_i with fixed multicast sessions, the sharing gain α for MSC_i increases (decreases). If the sharing gain α is used as a criterion, it is called *incremental change* when the sharing gain of a changed MSC is greater than η . Otherwise, it is called *decremental change*.

The dynamic behavior described above is for the OC sharing strategy. In the EC sharing strategy, any membership change of a multicast session in MSC_i is called *decremental change* since EC requires all multicast sessions in MSC_i must have the same membership. On the other hand, in the SC sharing strategy, if any membership change in MSC_i makes MSC_i unable to meet the SC criterion, it is called *decremental change*. Finally, if there is a membership change of a session in MSC_i with size 1, then it is always called *decremental change*.

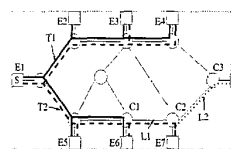


Figure 3: An example of link/core overlap in a Multicast Sharing Class.

3.2.1 Re-grooming Approaches with Dynamic Join/Leave

To re-groom existing MSCs differently or to groom unshared multicast sessions into MSCs with dynamic join/leave, we propose two approaches: time-based approach and need-based approach.

A. Time-based Approach

In the time-based approach, existing MSCs and unshared multicast sessions are evaluated at regular intervals. If there is a join/leave in the middle of interval, just update the membership information of the affected session(s) and the forwarding table of the affected MSC. If incremental change occurs in a MSC (say

MSC_i), the affected MSC_i is just updated for its membership and forwarding table. If decremental change occurs in the MSC_i , the MSC_i is marked as *changed* (MSC'_i) and its membership and forwarding table are updated. Then first calculate the sharing gain of the changed MSC'_i . If the sharing gain α is larger than η , just keep it as a new MSC. Otherwise, MSC'_i is evaluated with existing MSCs to see if it can fit in or added to one of them. A detailed description of a heuristic algorithm, called MEMBER-JOIN-LEAVE, is given in Appendix B. The heuristic algorithm is executed at the end of every interval. Note that the algorithm given in the Appendix B is for OC strategy. When the EC/SC strategy apply, it needs to check the EC/SC criterion in a similar heuristic to MEMBER-JOIN-LEAVE instead of calculating the sharing gain.

B. Need-based Approach

In the need-based approach, each MSC is evaluated only after there is a change in membership (link/core can be used.). If incremental change occurs in a MSC, just update its membership information and forwarding table. If decremental change occurs in a MSC (say MSC'_i), the MEMBER-JOIN-LEAVE is executed. Note that in the need-based approach there is only one changed MSC to check at a time because the MEMBER-JOIN-LEAVE is executed whenever a decremental change occurs. However, it is also possible to execute the MEMBER-JOIN-LEAVE algorithm after there are a certain number of membership changes above a threshold as a trade-off for computational and/or signaling overhead.

In the two approaches above, namely time-based approach and need-based approach, a MSC is not evaluated for re-grooming if incremental change occurs. This is because if incremental change occurs in a MSC, the MSC will provide better sharing gain than before. In addition, it aims to reduce the computational overhead from frequent re-grooming. As a matter of fact, one may try to evaluate sessions in U with the incrementally changed MSC, or to merge the changed MSC with other MSCs. However, this process will cause too much computational overhead per every interval or every change. Therefore a trade-off is needed. Instead, re-grooming is performed only when decremental change occurs.

Note that similarly as mentioned in birth/death operation, subsets of a changed MSC need to be evaluated with sessions in U because the subsets may not have been evaluated with each session in U previously according to the OC sharing strategy. In addition, newly formed MSCs by adding a session of a changed MSC need to be evaluated with sessions in U due to the same reason.

4 Simulation Results

4.1 Network Model and Assumptions

The network model used consists of edge routers and core routers. Each edge router is an access point to a backbone network consisting of core routers. For simulation, a random back-

bone network is generated. A more detailed description as to how the multicast and unicast traffic is generated, and how the multicast tree for each session is constructed is referred to [1].

To simulate dynamic sessions, initially there are a fixed number of multicast sessions, M , originating at the edge router and all edge routers have the same number of multicast sessions. Then events (creation or deletion of a session) occur according to a Poisson process with arrival rate μ at each edge router. Once a session is created, first its membership is determined and will remain the same until it expires.

In dynamic membership, all edge routers have the same number of multicast sessions, M , as in dynamic sessions and the number of multicast sessions will remain unchanged during simulation. To simulate dynamic membership, events (join or leave of a member in a session) occur according to a Poisson process with arrival rate μ at each edge router. Each event corresponds to a join or leave of a member. Then, first select one of multicast sessions at the edge router and decide one of members to leave (from existing ones) or a new member to join (that did not join a session yet.).

In addition to some of the parameters mentioned so far, there are a number of other parameters that affect the performance of various multicast schemes. The following are parameters examined and their default values. First the GB size is perhaps the important parameter affecting the performance of the various multicast schemes and its default size in the simulation is 4.5 percent of the average payload of unicast traffic. The burst assembly time for unicast/multicast traffic is 500 μ s and 1000 μ s, respectively. The number of core routers is 15 in the network. In addition, the amount of multicast traffic as a percentage of that of unicast traffic is 5 percent by default in the simulation and the initial number of multicast sessions M is 14. Finally the average number of members per multicast session is 70 percent of the total number of edge routers and the event arrival rate μ is 500, i.e., one event every 2000 μ s. Note that in the need-based approach, re-grooming is performed at every event.

4.2 Numerical Results

In this subsection, we present the simulation results. Whenever appropriate, we use the performance of S-MCAST as the base, that is, we determine the ratio of the average amount of multicast traffic per link using two multicast schemes (M-UCAST and TS-MCAST) over that using S-MCAST.

A. Effect of the number of core routers

Fig. 4 shows the effect of the network size (number of core routers) on the ratio of the average amount of multicast traffic per link obtained by M-UCAST and TS-MCAST to that obtained by the S-MCAST scheme. Note that in Fig. 4, M-UCAST is always worse than S-MCAST in the given network conditions (hence hereafter M-UCAST is not shown for comparison.). In addition, all TS-MCAST schemes (with EC, SC, OC, and OC-MAX) show a gradual decrease in the performance as the average path length (or number of hops) increases. Fig. 4 also shows

that the performance of TS-MCAST using OC strategy is very close each other. This is because when event occurs (dynamic sessions or dynamic membership), evaluation for re-grooming is based on the sharing gain α . Therefore the performance using other criteria than gain approaches that of gain criterion.

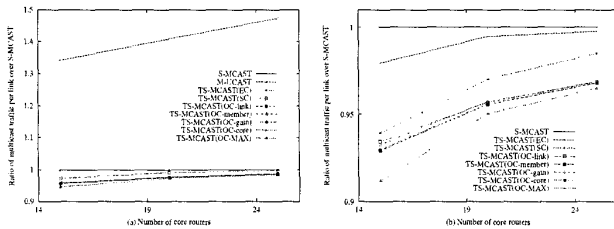


Figure 4: Effect of the number of core routers: (a) dynamic sessions and (b) dynamic membership - need-based approach.

B. Effect of the GB size

Note that the GB size is perhaps the most important parameter affecting the performance of various multicast schemes. Fig. 5 shows the effect of the GB size under dynamic sessions (a) and dynamic membership (b). Note that in the SC strategy gain check is not performed after forming MSCs, so the performance becomes worse than S-MCAST when the GB size is small while the number of control packets per link is reduced (to be shown in Fig. 7 (b)).

All OC criteria show similar performance each other due to the same reason described earlier. The OC-MAX strategy always outperforms OC strategy by paying the price for the computation time.

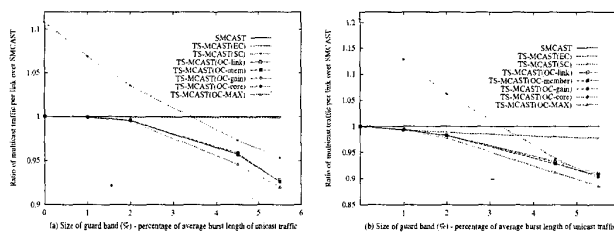


Figure 5: Effect of the GB size: (a) dynamic sessions and (b) dynamic membership - need-based approach.

C. Effect of the event arrival rate

Fig. 6 shows the relative performance when the event arrival rate increases from 100 to 500 per second (accordingly one event from every 10 burst assembly times to every 2 burst assembly times for multicast traffic on average). Note that there is no big performance degradation as the event arrival rate increases. This is because in the simulations the rate of birth/death (or join/leave) is the same on average. If the rate of birth/death (or join/leave) is not the same, the relative performance will increase or decrease considerably as the arrival rate changes. Note, however, that the performance under dynamic membership is more

affected positively than that under dynamic sessions in the given event arrival.

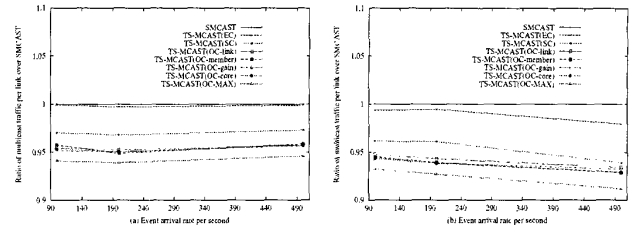


Figure 6: Effect of the event arrival rate: (a) dynamic sessions and (b) dynamic membership - need-based approach.

D. Effect of the re-grooming interval and on the number of control packets

Fig. 7 (a) shows the effect of the re-grooming interval on the relative performance under dynamic membership as the re-grooming interval increases from every 2 burst assembly times to every 100 burst assembly times. If the re-grooming interval is smaller than 20 burst assembly times ($20000 \text{ us} = 20 \text{ ms}$) for multicast traffic, the performance difference is very small between two approaches. However, if the re-grooming interval exceeds 20 burst assembly times of multicast traffic, the performance of time-based approach degrades linearly. Therefore, we observe from Fig. 7 (a) that a trade-off between the frequent re-grooming (thus computation and/or signaling overhead) and the performance of tree sharing is needed.

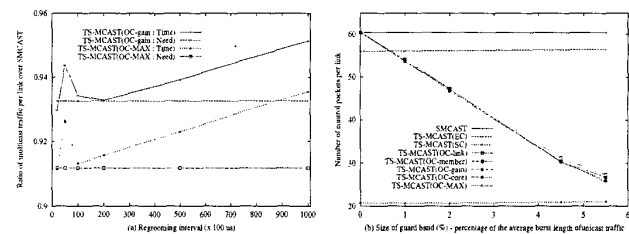


Figure 7: (a) Comparison of two approaches (time-based/need-based approach) and (b) the number of control packets per link with a different GB size (need-based approach).

Another benefit of tree sharing besides bandwidth saving is that it can reduce the number of control packets per link in the network, thus less processing load and smaller blocking probability. Fig. 7 (b) shows the number of control packets per link as the GB size increases. As the GB size increases, the more multicast sessions will share the GB, thus smaller number of control packets generated. The OC-gain and OC-MAX generate a smaller number of control packets than other criteria as the GB size increases. Note that the gain check is not performed in the EC/SC sharing strategies, thus the number of control packets per link from the SC (also EC with GB size 0) is smaller than

the OC/OC-MAX sharing strategies while SC consumes more bandwidth (see Fig. 5).

5 Conclusions

In this paper, we have studied three multicast schemes in optical burst-switched WDM networks under dynamic sessions and membership, namely S-MCAST, M-UCAST and TS-MCAST. The first one is natural and the other two are considered to reduce the amount of GBs (and the number of control packets) per unit of multicast data. We showed that the tree sharing under dynamic sessions and membership shows better performance than S-MCAST/M-UCAST as in static sessions and fixed membership [1].

To support tree sharing, we described several sharing strategies which can mix multiple multicast sessions to share the GB when assembling bursts. In addition, we proposed several heuristic algorithms which allow frequent re-grooming for supporting dynamic sessions and membership. Those heuristic algorithms perform well with less computational overhead when compared to OC-MAX strategy that is based on an exhaustive search so that the sharing gain at the edge router is maximized.

Appendix A

SESSION-DELETE Algorithm

S = set of MSCs (size of each ≥ 2); U = set of MSCs (size of each = 1);
 $H_i = S \cup U$;
 R = set of MSCs with size 1 after applying tree sharing strategy to a new MSC;
 P = set of MSCs (≥ 2) after applying tree sharing strategy to a new MSC;
 if a session $m \in MSC_k$ expires, $MSC_k = MSC_k - m$;
 if ($MSC_k == 1$)
 for each MSC_j ($j \neq k$) $\in S$,
 Compute_Gain($MSC_j = MSC_j \cup MSC_k$); // gain α_j
 if $\alpha_j > \eta$, keep it as a new MSC and
 evaluate each session $u \in U$ with the new MSC (MSC_j);
 add u to the new MSC only if u can fit in MSC_j ;
 end for
 if there is no $MSC_j \in S$ to add MSC_k , put MSC_k into U (becoming U') and
 apply tree sharing strategy using one of criteria to U' ;
 else
 Compute_Gain(MSC_k);
 if ($\alpha_k > \eta$) keep MSC_k as a new MSC;
 otherwise, apply tree sharing strategy using one of criteria to MSC_k ;
 $MSC_k = R \cup P$, $MSC_{kh} \in P$ and $h \leq |MSC_k|/2$;
 if ($R \neq \emptyset$)
 for each $r \in R$ evaluate it with $MSC_j \in (S - MSC_k)$;
 if r can fit in MSC_j (i.e., $\alpha_j > \eta$), add r to MSC_j ;
 otherwise (no MSC_j), put r into U (becoming U');
 if any, make all $MSC_{kh} \in P$ as new MSCs and
 evaluate $u \in U'$ with the new MSCs;
 apply tree sharing strategy to the remaining sessions
 in U' using one of criteria;
 else
 make $MSC_{kh} \in P$ as new MSCs and evaluate $u \in U$ with the new MSCs;
 end if
 end if

Appendix B

MEMBER-JOIN-LEAVE Algorithm

S = set of MSCs (size of each ≥ 2); U = set of MSCs (size of each = 1);
 $H_i = S \cup U$;

R = set of MSCs with size 1 after applying tree sharing strategy to a changed MSC;
 P = set of MSCs (≥ 2) after applying tree sharing strategy to a changed MSC;
 for each changed $MSC_j \in S$,

 if ($|MSC_j| > 1$),
 Compute_Gain(MSC_j); // gain α_j
 if $\alpha_j > \eta$, just keep MSC_j as a new MSC;
 then evaluate each session $u \in U$ with MSC_j ($MSC_j \cup u$) and
 add u to MSC_j only if MSC_j can provide gain $> \eta$;
 otherwise, apply tree sharing strategy using one of criteria and
 decompose MSC_j as $MSC_j = R \cup P$, $MSC_{jh} \in P$ and $h \leq |MSC_j|/2$;
 if ($R \neq \emptyset$)
 for each $r \in R$, evaluate it with $MSC_j \in (S - MSC_j)$ and add it to MSC_j
 if the sharing gain α_j of MSC_j ($MSC_j \cup r$) is greater than η ;
 otherwise, put it into U (becoming U') and evaluate each session $u \in U'$
 with all newly formed/changed MSCs (e.g., MSC_{jh} or MSC_j);
 if u can fit in one of the newly formed/changed MSCs, add it to that and
 apply tree sharing strategy to the remaining sessions in U
 using one of criteria selected;
 else
 evaluate each session $u \in U$ with all newly formed/changed MSCs
 (e.g., MSC_{jh} or MSC_j);
 if u can fit in one of the newly formed/changed MSCs, add it to that and
 apply tree sharing strategy to the remaining sessions in U
 using one of criteria selected;
 end if
 else // $|MSC_j| = 1$
 evaluate $MSC_j \in U$ with $MSC_k \in S$;
 Compute_Gain($MSC_j \cup MSC_k$); // gain α_k
 if $\alpha_k > \eta$, add MSC_j to MSC_k ;
 otherwise, apply tree sharing strategy to U using one of criteria;
 end if
 end for

References

- [1] M. Jeong, Y. Xiong, H. C. Cankaya, M. Vandenhoute and C. Qiao, "Efficient Multicast Schemes for Optical Burst-Switched WDM Networks," *Proc. of IEEE ICC 2000*, pp. 1289-1294, June 2000.
- [2] M. Jeong, C. Qiao, Y. Xiong and M. Vandenhoute, "Evaluation of Multicast Schemes in Optical Burst-Switched Networks - the Case with Dynamic Sessions," *Proc. of SPIE, Terabit Optical Networking*, vol. 4213, Nov. 2000.
- [3] C. Brackett, "Dense Wavelength Division Multiplexing Networks: Principles and Applications," *IEEE J. on Selected Areas in Communications*, pp. 373-380, Aug. 1990.
- [4] P. Green, *Fiber-Optic Communication Networks*. Prentice-hall, 1992.
- [5] R. Ramaswami, "Multi-wavelength Lightwave Networks for Computer Communication," *IEEE Communications Magazine*, no. 31, pp. 78-88, 1993.
- [6] B. S. Arnaud, "Architectural and Engineering Issues for Building an Optical Internet," *Proc. of SPIE, All-optical Networking*, vol. 3531, pp. 358-377, Nov. 1998.
- [7] C. Qiao and M. Yoo, "Optical Burst Switching (OBS) - A New Paradigm for an Optical Internet," *J. of High Speed Networks*, vol. 8, no. 1, pp. 69-84, 1999.
- [8] C. Qiao, M. Jeong, A. Guha, X. Zhang and J. Wei, "WDM Multicasting In IP over WDM Networks," *Proc. of IEEE ICNP '99 Proceedings*, pp. 89-96, Nov. 1999.
- [9] G. Sahin and M. Azizoglu, "Multicasting Routing and Wavelength Assignment in Wide-Area Networks," *Proc. of SPIE, All-Optical Networking*, vol. 3531, pp. 196-208, Nov. 1998.
- [10] L. H. Sahasrabudde and B. Mukherjee, "Light-Trees: Optical Multicasting for Improved Performance in Wavelength-Routed Networks," *IEEE Communications Magazine*, vol. 3, pp. 67-73, Feb. 1999.
- [11] R. Malli, X. Zhang and C. Qiao, "Benefits of Multicasting In All-Optical Networks," *Proc. of SPIE, All-Optical Networking*, pp. 209-220, Nov. 1998.
- [12] M. Yoo, M. Jeong and C. Qiao, "High-Speed Protocol for Bursty Traffic in Optical Networks," *Proc. of SPIE, All-Optical Communication Systems*, vol. 3230, pp. 79-90, Nov. 1997.
- [13] X. Zhang, J. Wei and C. Qiao, "On Fundamental Issues in IP Over WDM Multicasting," *Proc. of IEEE Int'l Conf on Comp. Comm. and Networks (IC3N)*, pp. 84-90, Oct. 1999.
- [14] D. Bertsekas and R. Gallager, *Data Networks*. Prentice-hall, 1992.
- [15] H. Takahashi and A. Matsuyama, "An Approximate Solution for the Steiner Problem in Graphs," *Math Japonica*, vol. 24, no. 6, pp. 573-577, 1980.