Clustering Based Scheduling: A New Approach to the Design of Scheduling Algorithms for WDM Star Networks

Sophia G. Petridou, Panagiotis G. Sarigiannidis, *Graduate Member, IEEE*
Georgios I. Papadimitriou, *Senior Member, IEEE*, Andreas S. Pomportsis
Aristotle University
email:{spetrido,sarpan,gp,apombo}@csd.auth.gr

***Abstract*— Scheduling algorithms in Wavelength Division Multiplexing (WDM) single hope networks aim at producing an effective schedule in order to improve the networks' performance. Up to now, popular approaches schedule network traffic based on nodes' requests which are considered in a sequential service order. This paper presents a novel packet scheduling scheme for WDM star networks based on clustering techniques. Our Clustering Based Scheduling Algorithm (CBSA) organizes the nodes of a network into groups (i.e. clusters) according to the number of their requests per channel and then it defines their transmission priority beginning from the nodes belonging to the cluster with greater demands and ending to the nodes of cluster with fewer requests. The simulation results have shown that the proposed approach improves network performance since it results in higher network throughput keeping mean packet delay at low levels in comparison with conventional scheduling algorithms.**

I. Introduction

Popular approaches schedule traffic in local area Wavelength Division Multiplexing (WDM) single hope networks [1] considering nodes in a sequential service order [2], [3]. However, sequential scheduling leads to a significant performance degradation in terms of network throughput and mean packet delay. This paper presents a new pre-transmission coordination scheduling algorithm for optical WDM star networks based on the clustering [4], [5] of network nodes. The proposed Clustering Based Scheduling Algorithm (CBSA) rearranges the service order of the network nodes by organizing them into groups (i.e. clusters) according to the number of their requests per channel. Then, our algorithm defines their transmission priority beginning from the nodes belonging to the cluster with greater demands and ending to the nodes of cluster with fewer requests. In this way, we decrease both the unused timeslots as well as the schedule length and as a result the network performance is significantly upgraded.

This work is inspired by the fact that a sequential service order scheme has the drawback of scheduling nodes without taking into account their specific requests. As a result, nodes with short length requests (few packets) may transmit prior to those with long length requests leading to decreased channel utilization because of many unused timeslots. Thus, it is important to rearrange the nodes' service order according to their requests on each channel. Therefore, a nodes' clustering approach is introduced which could create groups of nodes with similar request patterns. Discovering such similar patterns and prioritizing them properly could lead to higher network performance without aggravating the time complexity of the scheduling algorithm. Clustering has already been used in many domains and especially on the Web aiming at improving Web applications [5], [6].

The remainder of this paper is organized as follows. Section II provides the network structure while Section III presents related packet scheduling algorithms for WDM star networks. Clustering background is given in Section IV. Section V presents our new scheduling algorithm while Section VI discusses the simulation results. Finally, conclusions and future work insights are given in Section VII.

II. Network Structure

A local area WDM single-hop network with broadcast and select architecture is considered. This network consists of n nodes, which are connected in a passive star coupler via a two-way optical fiber, and w channels (wavelengths), where $n \geq w$. Each node may transmit packets on different channels using a tunable transmitter, while it receives packets in a dedicated channel (home channel) using a fixed receiver, as depicted in Fig. 1. In this TT-FR implementation, it is clear that two or more nodes may transmit on the same channel at the same time causing channel collision. Thus, a media access protocol (MAC) is needed to support a set of access rules aiming at preventing collisions and specifying the way that nodes transmit on the available channels [7], [8].

The proposed scheduling algorithm is based on global status information which, in our case, is the n x w demand matrix D, where $d(i, j)$ element, $i = 1, \ldots, n$ and $j = 1, \ldots, w$, indicates the number of data packets at node u_i that are destined to channel λ_j. Time is divided in timeslots with the packet transmission time to be equal to one timeslot while the transmission process is organized in transmission frames. Each frame has two phases namely the reservation and the data phase. During the reservation phase the n nodes send their requests to the common data channels which are then recorded in the demand matrix D. At the same time, the CBSA operates in conjunction with a distributed scheduling

1-4244-1370-2/07/$25.00 ©2007 IEEE

Symbol	Definition
n, w	Number of nodes and channels
$U = \{u_1, \ldots, u_n\}$	The set of network's nodes
$\Lambda = \{\lambda_1, \ldots, \lambda_w\}$	The set of network's channels
D	The n x w demand matrix
k	The upper bound of nodes' requests
t	Schedule's length in timeslots
$L = \{l_1, \ldots, l_t\}$	The set of timeslots
S	The w x t scheduling matrix
Cl	Clustering process
noc	Number of clusters
C_j	Cluster, $j = 1, \ldots, noc$
$f(u_i, C_j)$	Function membership of node u_i to cluster C_j, $i = 1, \ldots, n$
c_j	Cluster representative
MeansD	The noc x w clusters representatives' demand matrix
$dist$	Nodes distance over channels
J	Criterion function

TABLE I

BASIC SYMBOLS NOTATION

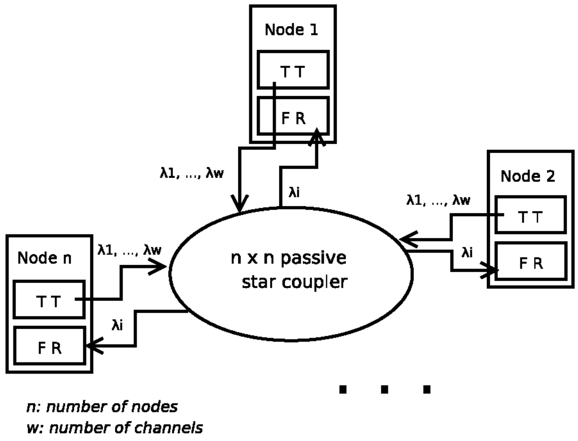


Fig. 1. Network Topology

algorithm and produces the w x t scheduling matrix S, where t denotes the length of the schedule in timeslots. Each $s(i, j)$ element, $i = 1, \ldots, w$ and $j = 1, \ldots, t$, represents the node that transmits on channel λ_i during the timeslot l_j.

III. RELATED SCHEDULING ALGORITHMS

A typical pre-transmission, coordination based scheduling algorithm for optical WDM networks is the Online Interval-based Scheduling (OIS) [2]. OIS incorporates on-line scheduling and has low time complexity $O(nw^2k)$, where k is the upper bound of nodes' requests on each channel. According to this algorithm each node needs to maintain a list of time intervals that are available on every data channel. Furthermore, for each node whose request is being processed, nodes maintain one additional list of intervals that have not yet been assigned to the specific node for transmission. These intervals show the unallocated time on a specific channel or node. More specifically, whenever a node has a request on a channel the algorithm examines the available intervals on this channel. Furthermore, the node's list of interval is also checked to determine if the node is scheduled to transmit on any other channel during the time interval requested. It is apparent that the algorithm does not assign more than one node at the same interval for the same channel in order to keep the schedule collision free. Hence, the transmission is eventually scheduled to the appropriate intervals and the lists are updated.

An extension of OIS, which is based on traffic prediction, is the Predictive Online Scheduling Algorithm (POSA) [3]. POSA has a very effective traffic prediction mechanism aiming at drastically reducing the computation time of the schedule. This mechanism differentiates POSA from OIS while both of them operates the same scheduling algorithm. More specifically, POSA maintains a set of n x w predictors which try to predict the demand matrix D for the next frame according to the history of the recent requests. Then it transmits node's requests according to these predictions. At the same time, it records the actual requests of the current frame to the predictors' history queues. Thus, this scheme saves valuable time since the scheduling algorithm allows the packets' transmission in parallel with the prediction of the next transmission requests. As a result, POSA leads to a significant decrease of the schedule estimation time.

IV. NETWORK NODES CLUSTERING

Under a particular Cl clustering process noc denotes the number of clusters to be created. Then, U denotes the set of nodes $U = \{u_1, \ldots, u_n\}$ that is to be clustered while $C_1, \ldots, C_{noc}$ denote each of the noc clusters consisting of $|C_1|, \ldots, |C_{noc}|$ members (i.e. nodes). Under this notation, the clustering process Cl is defined as the assignment of network nodes to groups of nodes (i.e. clusters):

$$Cl : \{1, \ldots, n\} \longrightarrow \{1, \ldots, noc\}$$

Nodes assigned to the same cluster are "similar" to each other and "dissimilar" to the nodes belonging to other clusters in terms of their packets requests per channel. The membership of a node u_i, where $i = 1, \ldots, n$, to a cluster C_j, where $j = 1, \ldots, noc$, is defined by the function f as follows:

$$f(u_i, C_j) = \begin{cases} 1 & if u_i \in C_j \\ 0 & otherwise \end{cases}$$

Based on the above, it is apparent that the notion of similarity is fundamental in a clustering process, and so far it is quite common to evaluate the dissimilarity between two items (in our case nodes) by using a distance measure [4]. To proceed with our network nodes' clustering process, we employ the Squared Euclidean distance[1] which is a well-known and widely used distance measure in the vector-space

[1]The Squared Euclidean distance uses the same equation as the Euclidean distance, but does not take the square root. For two points $P = (p_1, \cdots, p_n)$ and $Q = (q_1, \cdots, q_n)$ in n-space their Squared Euclidean distance is defined as: $\|p_i - q_i\|^2$

model [4], [6]. Given that each node u_i is represented by the row i of the demand matrix D, this row is denoted as a multivariate vector consisting of w values as follows:

$$D(i,:) = (d(i,1), \ldots, d(i,w))$$

Therefore, the evaluation of the dissimilarity between two nodes can be expressed by the distance of their demand vectors. Thus, we will use the expression $dist(u_x, u_y)$, where $u_x, u_y \in U$, to denote the Squared Euclidean distance of the nodes' demand vectors $D(x,:)$ and $D(y,:)$:

$$dist(u_x, u_y) = \|D(x,:) - D(y,:)\|^2$$

Then, an arbitrary cluster C_j, where $j = 1, \ldots, noc$, of the nodes' set U is considered. The representation of the cluster C_j when clustering process Cl is applied to it, collapses the nodes belonging to C_j into a single point (e.g. the mean value which does not correspond to an existing node). This point is called cluster's representative c_j (also known as centroid) since each node $u_i \in C_j$ is represented by c_j. Given the demand vectors of $u_i \in C_j$, the demand vector of c_j is defined as follows:

$$\mathit{MeansD}(j,:) = \frac{1}{|C_j|} \sum_{i=1}^{n} f(u_i, C_j) * D(i,:), j = 1, \ldots, noc$$

Since both $D(i,:)$ and $\mathit{MeansD}(j,:)$ are vectors, their dissimilarity is measured by their Squared Euclidean distance $dist(u_i, c_j)$. Considering all clusters, the clustering process is guided by the *criterion function* J which is defined to be the sum of distances over all channels between each node and the representative of the cluster that the node is assigned to:

$$J = \sum_{j=1}^{noc} \sum_{u_i \in C_j} dist(u_i, c_j)$$

Based on the above we can define the network nodes clustering as follows: Given a network with n nodes whose packets' requests on each of the w channels are organized in an n x w demand matrix D, the integers noc and k, and the criterion function J, find a Cl clustering of U into noc clusters such that the J is minimized.

V. THE PROPOSED CLUSTERING BASED SCHEDULING ALGORITHM

The CBSA is a two-step process depicted in Fig. 2. Its core idea is that network nodes should be rearranged according to their packets' requests before their final schedule. During the first step, the Cl clustering of the nodes' set U is produced based on the n x w demand matrix D. For this clustering the K-means algorithm is employed which is a widely used partitional clustering algorithm [9]. The K-means minimizes the objective function J defined in Section IV. Next, given the Cl as well as the *MeansD* table, consisting of the clusters representatives' demand vectors $\mathit{MeansD}(j,:)$, the *SortedMeans* is computed in order that we prioritize the clusters with greater requests. The calculated *SortedMeans* is then used in order that the network nodes are rearranged. Once the *ClusteredNodes* is formed, the algorithm proceeds to the second step called the scheduling step. The goal of function *Schedule* is to form the scheduling matrix S using the same logic as POSA algorithm.

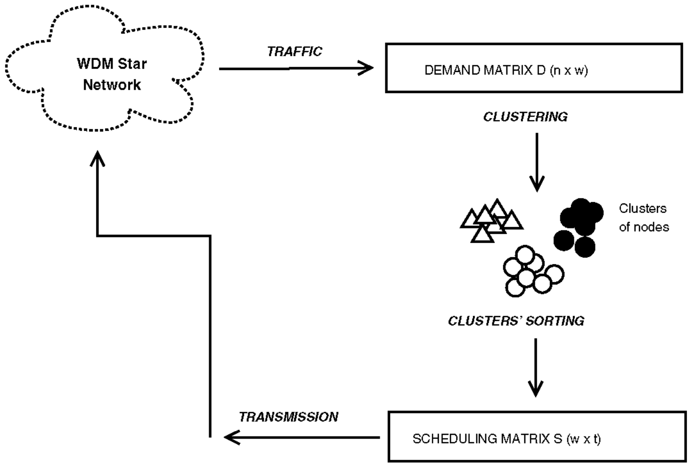


Fig. 2. The CBSA overview

Algorithm 1 The CBSA flow control

Input: A set U of n nodes whose packets' requests on each of the w channels organized in an n x w demand matrix D, the upper bound on nodes' requests k and the number of clusters noc.

Ouput: The scheduling matrix S.

1: /*Clustering Step*/
2: $(Cl, MeansD) = K - means(D, noc)$
3: $SortedMeans = \mathit{Quicksort}(MeansD)$
4: $ClusteredNodes = \mathit{Arrange}(SortedMeans)$
5: /*Scheduling Step*/
6: $S = \mathit{Schedule}(ClusteredNodes)$

Theorem 1: The CBSA has time complexity $O(nkw^2)$.

Proof: During the clustering step we employ the K-means algorithm (line 2) whose time complexity is $O(n\ noc\ r)$, where n is the number of nodes, noc the number of clusters to be created and r the number of iterations that takes the algorithm to converge. However, both noc and r are relatively small compared to the number of nodes n and thus their contribution to the algorithm's complexity can be ignored [4]. Thus, the Cl clustering is computed in time linear on the number of nodes: $O(n)$. The *Quicksort* function (line 3) sorts clusters' representatives in $O(noc\ log(noc))$ time while the *Arrange* function (line 4) takes time $O(n)$ to arrange the n nodes according to the *SortedMeans*. The total time complexity of the clustering step is thus $O(n+noc\ log(noc)+n)$ which becomes $O(n)$. During the scheduling step the *Schedule* function (line 5) needs $O(nkw^2)$ time [3] to from the scheduling matrix S, where k is the upper bound of nodes' requests and w the number of channels. As a result, the total complexity of CBSA is $O(n) + O(nkw^2) = O(nkw^2)$. ■

To facilitate the comprehension of the CBSA, consider a network consisting of $n = 4$ nodes namely $U =$

$\{u_1, u_2, u_3, u_4\}$ and $w = 3$ channels namely $\Lambda = \{\lambda_1, \lambda_2, \lambda_3\}$ while the upper bound of nodes' requests is $k = 3$. Then, a 4 x 3 demand matrix D could be the following:

$$D = \begin{pmatrix} 0 & 1 & 2 \\ 2 & 1 & 0 \\ 1 & 3 & 2 \\ 2 & 3 & 3 \end{pmatrix}$$

Example 1. In the above demand matrix D the fact that $D(3,2) = 3$ means that the node identified as u_3 requests 3 packets on channel λ_2. □

Applying the K-means for $noc = 2$ in the above D matrix results in $Cl = (2,2,1,1)$ which means that $u_1, u_2 \in C_2$ while $u_3, u_4 \in C_1$. Given this Cl the clusters representatives' demand matrix *MeansD* is formed as follows:

$$MeansD = \begin{pmatrix} 1.5 & 3 & 2.5 \\ 1 & 1 & 1 \end{pmatrix}$$

Sorting *MeansD* results in giving priority to C_1 and thus to nodes u_3, u_4. Therefore, the schedule service order defined by the CBSA will be u_3, u_4, u_1, u_2 instead of u_1, u_2, u_3, u_4 which is formed by the POSA. Tables II and III depict the scheduling matrix S produced respectively. Based on these tables, CBSA provides 16.7% improvement on channels' utilization while it reduces the mean packet delay from 4.6 to 3.7 timeslots.

	Timeslots									
	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}
w_1	u_3	u_4	u_4	u_2	u_2					
w_2	u_1	u_3	u_3	u_3	u_4	u_4	u_4	u_2		
w_3		u_1	u_1		u_3	u_3		u_4	u_4	u_4

TABLE II

THE SCHEDULING MATRIX S PRODUCED BY CBSA

	Timeslots											
	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}	l_{11}	l_{12}
w_1	u_2	u_2	u_3	u_4	u_4							
w_2	u_1		u_2	u_3	u_3	u_3	u_4	u_4	u_4			
w_3		u_1	u_1				u_3	u_3		u_4	u_4	u_4

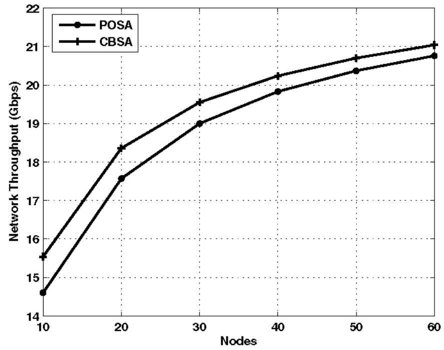
TABLE III

THE SCHEDULING MATRIX S PRODUCED BY POSA

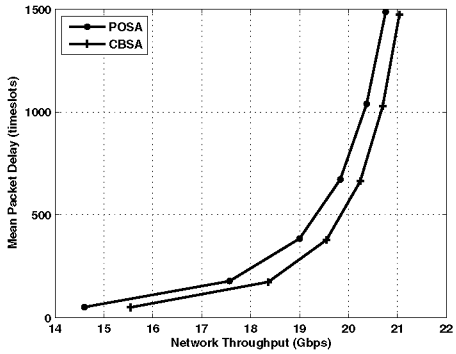
VI. SIMULATION RESULTS

To evaluate the performance of the CBSA we compared it with the POSA. The experiments we carried out are based on the following assumptions:

1) Traffic pattern is uniform i.e. packets' requests are generated with equal probability for every node.
2) Nodes may request 0 to k packets on each frame with equal probability.
3) The line is defined at 3 Gbps per channel and the tuning time is considered to be negligible.
4) The outcome results from 10000 transmission frames.



(a) Network throughput as a function of the number of nodes

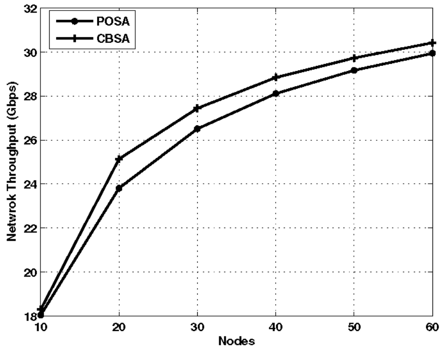


(b) Mean packet delay as a function of the network throughput

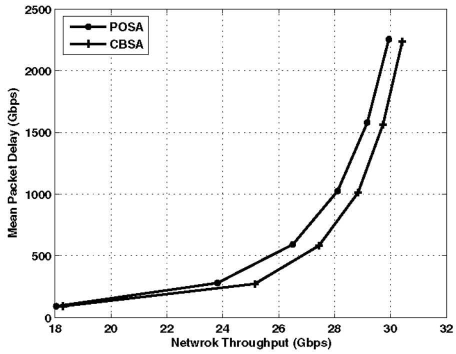
Fig. 3. Results for $w = 8$ and $noc = 7$

In the simulation results shown in this section, the performance of POSA and CBSA is presented in terms of network throughput and mean packet delay. Network throughput represents the average number of bits transmitted per frame on each channel while mean packet delay denotes the mean time in timeslots that packets are waiting at the queues till the beginning of their transmission. We experimented with different number of nodes (n) and channels (w).

In Fig. 3(a) and 4(a) the algorithms' input is set to $n = 10, 20, 30, 40, 50, 60$ nodes while the number of channels are $w = 8$ and $w = 12$ respectively. Moreover, we fixed the upper bound of nodes' requests in $k = \lceil (n*w)/5 \rceil$ [3] for scalability reasons and set the number of clusters to $noc = 7$. From Fig. 3(a) and 4(a), which depict the network's throughput as a function of the number of nodes, it is clear that CBSA is steadily superior to POSA both for $w = 8$ and $w = 12$. This is expected since CBSA prioritizes the long length requests and thus allocates more free timeslots for the rest requests. The maximum and minimum observed differences are 1.33 Gbps (for $n = 20$ nodes and $w = 12$ channels) and 0.25 Gbps (for $n = 10$ nodes and $w = 12$ channels) correspondingly. Fig. 3(b) and 4(b), which represent mean packet delay as a function of the network throughput, validate the CBSA's superiority since it is obvious that the improvement in network's throughput



(a) Network throughput as a function of the number of nodes



(b) Mean packet delay as a function of the network throughput

Fig. 4. Results for $w = 12$ and $noc = 7$

does not affect the mean packet delay. More specifically, the CBSA keeps lower the mean packet delay in comparison with POSA independently of the number of channels while obtains higher network throughput. For example, for $n = 30$ and $w = 12$ CBSA offers 27.44 Gbps network throughput causing 583 timeslots as mean packet delay while the respective values for POSA are 26.51 Gbps and 594 timeslots.

VII. Conclusions and Future Work

This paper introduces and evaluates a novel packet scheduling algorithm for WDM star networks. The proposed Clustering Based Scheduling Algorithm (CBSA) rearranges the service order of a network's nodes by organizing them into groups (i.e. clusters) according to the number of their packet requests per channel. Then it defines the nodes' transmission priority beginning from the cluster with greater demands and ending to the cluster with fewer requests. The proposed algorithm has been evaluated under uniform traffic for different number of nodes and channels and it has resulted in upgrading the network performance while keeping low the mean packet delay in comparison with the POSA.

Future work aims at evaluating the proposed scheme under poisson and bursty traffic. Moreover, the experimental results offer insight for sorting not only the clusters but also, at a second step, the nodes (members) in each cluster according to their packet transmission requests. Handling appropriately the nodes in each cluster is expected to further improve the network's performance.

References

[1] B. Mukherjee, *Optical WDM Networks*. Springer, 2006.

[2] K. Sivalingam, J. Wang, J. Wu, and M. Mishra, "An intervalbased scheduling algorithm for optical wdm star networks," *J. Photonic Network Commun.*, vol. 4, no. 1, pp. 73–87, 2002.

[3] E. Johnson, M. Mishra, and K. Sivalingam, "Scheduling in optical wdm networks using hidden markov chain based traffic prediction," *J. Photonic Network Commun.*, vol. 3, no. 3, pp. 271–286, 2001.

[4] A. Jain, M. Murty, and P. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[5] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan, "Web usage mining: Discovery and applications of usage patterns from web data," *SIGKDD Explorations*, vol. 1, no. 2, pp. 12–23, 2000.

[6] S. Petridou, V. Koutsonikola, A. Vakali, and G. Papadimitriou, "A divergence-oriented approach for web users clustering," in *Proc. of International Conference on Computational Science and its Apllications (ICCSA '06)*, Glaskow, Scotland, May 2006, pp. 1229–1238.

[7] G. Papadimitriou, P. Tsimoulas, M. Obaidat, and A. Pomportsis, *Multi-wavelength Optical LANs*. Wiley, 2003.

[8] P. Sarigiannidis, G. Papadimitriou, and A. Pomportsis, "A high throughput scheduling technique, with idle timeslot elimination mechanism," *IEEE Journal of Lightwave Technology*, vol. 24, no. 12, pp. 4811–4827, 2006.

[9] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.