Adaptive Open-Shop Scheduling for Optical Interconnection Networks

Dung Pham Van, Matteo Fiorani, Lena Wosinska, and Jiajia Chen

***Abstract*—This paper deals with resource management in optical interconnection networks. It first proposes an optical resource management framework as a platform to develop and evaluate efficient solutions for multipoint-to-multipoint optical communication systems with a centralized controller. The paper then focuses on studying the optical resource scheduling (ORS) problem as a core element in the framework by applying the classical open-shop scheduling theory. The ORS problem can therefore be solved by adopting the existing preemptive and non-preemptive open-shop scheduling algorithms. In an optical network with non-negligible reconfiguration delay, a preemptive algorithm may incur high reconfiguration overhead resulting in worse performance compared to the non-preemptive strategy. Motivated by this fact, this paper proposes an adaptive open-shop scheduling algorithm (AOS) that dynamically decides the optimal scheduling strategy according to traffic condition and system parameters, such as reconfiguration delay, non-preemptive approximation ratio, and number of involved optical interfaces. The solution is assessed by means of an analytical model that allows to quantify the network performance in terms of packet delay and potential energy savings obtained by the sleep mode operation. As a possible application scenario, the inter- and intra-rack optical interconnection networks in data centres are considered. Analytical results demonstrate how the proposed AOS outperforms the non-preemptive and preemptive scheduling strategies for typical configurations used in data centre networks. In addition, the reconfiguration delay and wake-up time of optical devices are identified as performance-determining factors.**

***Index Terms*—Data centre networks, open-shop scheduling, optical interconnects, performance analysis, resource management.**

I. Introduction

The performance of data centre (DC) interconnects is gaining intense interests as cloud based applications span over a growing number of servers rising traffic volumes and risk for network congestion [1]. It has been reported that DC traffic is increasing at a very high compound annual growth rate (CAGR) of 25% [2], of which the majority is exchanged among servers within the same DC facility [3]. To cope with major limitations in scalability and energy efficiency introduced by current electrical packet switching-based DC networks (DCNs), several architectural solutions for optical DCNs have recently been presented, e.g., [4]–[7]. Provided that the majority of links and switching devices in future DCNs might be based on optical technologies, it is of vital importance to have efficient mechanisms for dynamic optical resource management in order to unleash the full potential of the optical interconnect technologies [1]. Such mechanisms should be able to optimize the performance while taking into account various aspects, such as traffic condition and reconfiguration delay, which is non-negligible in realistic communication systems.

This paper first proposes a centralized optical resource management (CORM) framework for optical interconnection networks, which are characterized by a multipoint-to-multipoint transmission in which all network nodes can transmit/receive messages to/from each other with a centralized controller and non-negligible reconfiguration delay of optical devices. The proposed CORM aims at providing a flexible platform in order to facilitate the development and assessment of efficient resource management schemes. In particular, it defines a communications protocol, relevant signaling messages, as well as a mechanism to improve the overall energy efficiency with sleep mode scheduling. The paper then pays particular attention to the optical resource scheduling (ORS) problem by applying the classical scheduling theory. In particular, the ORS problem is formulated as an open-shop scheduling problem, in which a number of jobs in a "shop" are scheduled to be processed by a set of machines with the objective to minimize the total processing time. Mapping the ORS to the open-shop scheduling problem allows leveraging on the existing scheduling strategies, algorithms, and mathematical derivations for developing and evaluating new solutions. The open-shop problem has two categories, i.e., preemptive and non-preemptive [8]. Preemption means that a transmission (job) can be divided into multiple sub-transmissions. Preemptive scheduling algorithms are characterized by the number of reconfigurations required to accomplish all the sub-transmissions.

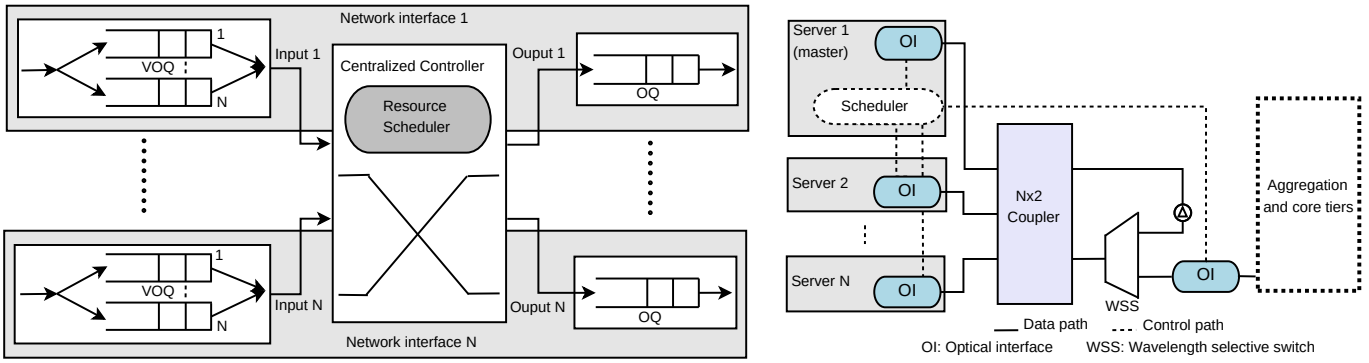
There exist many algorithms for both preemptive and non-preemptive open-shop scheduling strategies, developed in different contexts such as satellite-switched time-division multiple access (SS/TDMA) [10], optical switches [11]–[13], and optical access networks [14]–[16]. However, when reconfiguration delay is not negligible, it has not been previously shown which strategy (preemptive or non-preemptive) provides better performance than the other. Furthermore, to the best of authors' knowledge, most existing studies consider simultaneous reconfiguration, in which interfaces cannot be reconfigured while others are transmiting/receiving traffic. In many cases, arbitrary reconfiguration is allowed in communication systems, and thus more advanced scheduling algorithms are of high

A short version of this work and preliminary results has been presented at IEEE Global Communications Conference (GLOBECOM), December 2016, Washington, DC USA.

Dung Pham Van and Matteo Fiorani are with Ericsson Research, Kista, 164 80 Stockholm, Sweden. Email: {dung.pham.van, matteo.fiorani}@ericsson.com.

Lena Wosinska and Jiajia Chen are with the Optical Networks Lab (ONLab), KTH Royal Institute of Technology, 164 40 Stockholm, Sweden. Email: {wosinska, jiajiac}@kth.se. Jiajia Chen is the corresponding author.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2017.2695078, Journal of Lightwave Technology



(a) Logical structure of the considered architecture

(b) An example of small-scale DCN architecture for intra-rack communications [9]

Fig. 1. Generic architecture and its example in optical data centre networks.

importance. This paper studies the open-shop scheduling problem in the context of high-performance optical interconnect networks for optical DCs, where reconfiguration is performed in an arbitrary manner with a non-negligible delay. To this end, the paper proposes an adaptive open-shop scheduling (AOS) algorithm that aims to optimize the scheduling based on well-known preemptive and non-preemptive algorithms while taking into account the non-negligible reconfiguration delay, traffic condition, and system configurations. The CORM framework and AOS are evaluated by means of an analytical model in the context of optical DC interconnection networks.

This work is an extension of the initial study presented in [17]. The major contributions of the paper are threefold. First, it introduces a flexible protocol framework that facilitates the development and assessment of different resource management solutions for multipoint-to-multipoint optical communication systems. Second, as the core element of the framework, the ORS problem is identified as an open-shop scheduling problem that allows to develop efficient resource allocation methods leveraging on the existing strategies. As a result, the AOS algorithm is proposed to adaptively optimize the scheduling performance by taking into account various factors, such as traffic condition and reconfiguration delay. Furthermore, for the first time, this paper presents an analytical model that is able to translate the makespan obtained by the open-shop algorithms into the packet delay and quantify the trade-off between the allowable packet delay and energy saving gain from sleep mode implementation.

II. Centralized Optical Resource Management

This section describes the considered optical interconnect system architecture for DCNs and presents the proposed CORM framework by detailing its operation, signaling mechanism, and opportunity for sleep mode implementation.

A. Considered System and Application Scenario

The multipoint-to-multipoint interconnection network architecture with centralized control can be considered as a crossbar switch, as illustrated in Fig. 1(a). The system consists of N network interfaces, each featuring an input port and an output port. To eliminate head-of-line (HOL) blocking, each network interface at an input port maintains a separate virtual output queue (VOQ) for each of N destined interfaces, whereas a common output queue (OQ) is used for all incoming traffic from other interfaces [18]. The interfaces are interconnected via a centralized controller that features a resource scheduling algorithm. The scheduler decides which inputs will be connected to which outputs for a certain duration (time slot) in a working cycle. At the beginning of a cycle, the scheduler examines the contents of the N^2 input queues and determines a conflict-free match between inputs and outputs with the minimal total time required for transmitting all traffic in the queues. In the considered architecture, the centralized controlling is employed, which helps reduce communication overhead as well as design complexity compared to its counterpart, namely distributed management approach [19].

The considered system can find its application in optical DCNs, both at the inter-rack and intra-rack levels, i.e., interconnection among racks and communications between servers within a rack, respectively. More specifically, inter-rack interconnect architectures, e.g., [4] are usually based on optical circuit switching, in which a micro-electro-mechanical system (MEMS) based optical core switch interconnects different top-of rack (ToR) switches to guarantee any-to-any connectivity. Note that ToR switches may be packet routers with optical-electrical-optical (OEO) converter and buffering capabilities or based on passive optical architecture, e.g., [9]. Thus, ToR switches play the role of network interfaces, whereas the optical core switch employs a centralized controller to perform the resource allocation. The logical structure is shown in Fig. 1(a). It is worth mentioning that even though simultaneous reconfiguration prevails in many optical switch architectures, with technology advance, ports of an optical core switch circuit can be reconfigured while others are transmitting/receiving traffic, i.e., allowing for arbitrarily reconfigurable capability [11]. In the case of intra-rack communications, a typical CORM-like intra-rack architecture was recently introduced in [9], where servers in a rack are interconnected by means of optical interconnect devices such as passive optical couplers and wavelength selective switch, as depicted in Fig. 1(b). In such optical ToR architecture, each optical network interface (OI) is equipped with a tunable transceiver to transmit (receive) traffic to (from) other interfaces in the rack, where each transmission takes place on a separate wavelength. Thus, the transmitter and receiver of an interface play the role of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2017.2695078, Journal of Lightwave Technology

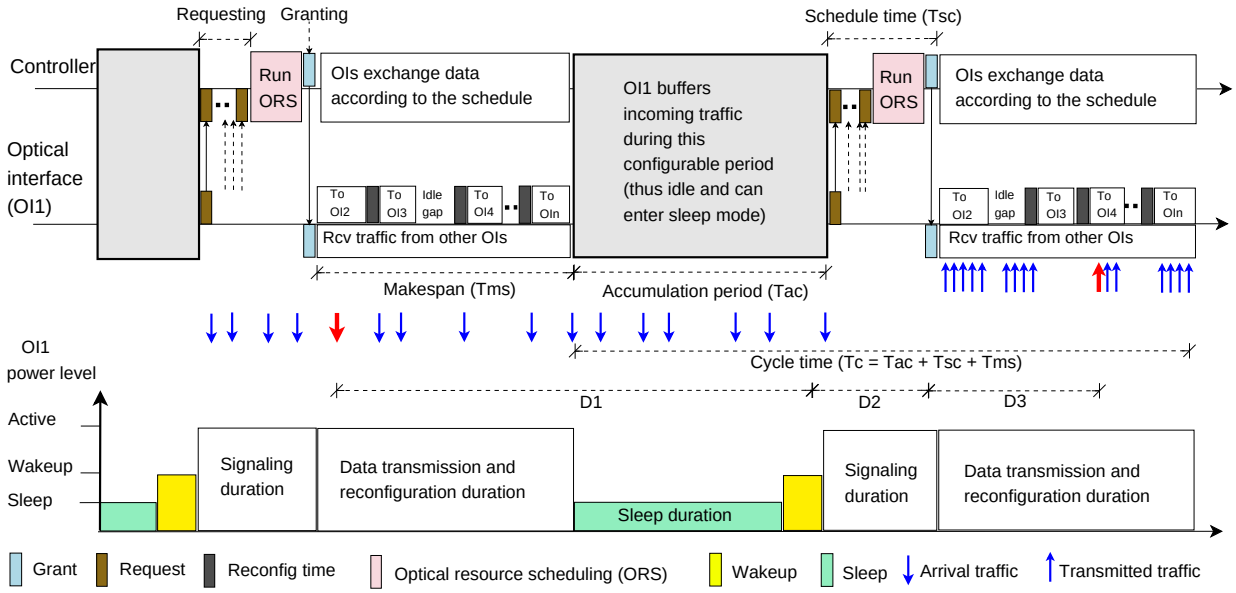


Fig. 2. Centralized optical resource management (CORM) framework: Operation (upper part) and Power consumption model (lower part).

the input and output port in Fig. 1(a), respectively. Each optical transceiver can tune to a specific wavelength for data transmission in an arbitrary (independent) manner.

B. CORM Operation and Signaling Mechanism

The CORM framework defines a signaling mechanism between the controller and OIs. In CORM, the controller follows the offline scheduling approach, that is, it waits for all the requests from OIs before scheduling. In this way, the controller can have a complete input traffic matrix of all network nodes and thus run the algorithm only one time for a working cycle.

The operation principle of the proposed CORM framework is illustrated in Fig. 2. For simplicity, only optical interface 1 (OI1) is shown. The timeline of the controller and the OI is composed of working cycles, each consisting of an accumulation period T_{ac}, a scheduling period T_{sc}, and a duration for data transmissions and reconfigurations T_{ms}. During the T_{ac} period, traffic is accumulated and the OIs are idle. During the scheduling period T_{sc}, the centralized controller collects information from its OIs, builds up a traffic matrix, runs an ORS algorithm for the input matrix, and notifies the OIs of the schedule computed by the ORS. During the T_{ms} period, conflict-free parallel data transmissions take place, in which a connection between an input and output ports is established for a certain period of time (slots). Upon expiration of the transmission duration, the ports are reconfigured to connect to other ports for new transmissions. This reconfiguration may take non-negligible time to be completed. After the T_{ms} duration, a new cycle starts and all the aforementioned steps are repeated. Note that in the CORM framework packets are buffered at network nodes, each being assumed to be equipped with N VOQs for N respective OIs (see also Fig. 1(a)).

In the CORM, the accumulation period T_{ac} is introduced to provide operators with more flexibility to adjust the network performance according to the users' requirements. In many cases, in a cycle, the actual data transmission is rather short with respect to the protocol overhead including the reconfiguration delay and signaling time. As a result, OIs are too busy for reconfiguring and exchanging control messages with the controller rather than for actual data transmissions. On the other hand, T_{ac} incurs additional waiting time of a packet. The larger the T_{ac}, the higher the packet delay. Therefore, T_{ac} should be configured properly to avoid high protocol overhead, while guaranteeing users' requirements on quality of service (QoS).

In general, the signaling mechanism can be designed in either out-of-band or in-band fashion. The former, such as the one presented in [9] helps reduce signaling overhead since data transmissions can take place during the scheduling period (see Fig. 2), yet requires an additional transceiver for each OI to realize the separate signaling channel. On the contrary, the latter approach uses the same channel for both control signaling and data transmission, and thus does not require the extra costs for the additional transceivers. This paper considers the in-band signaling method while leaving the in-band vs. out-of-band signaling comparison for future investigation. Note that the scheduler may be hosted by a specific network node, i.e., the master server in intra-rack DCNs, as illustrated in Fig. 1(b).

In the CORM, OIs report their traffic demands to the controller by means of `Request` messages, similar to the reporting process in Ethernet Passive Optical Networks (EPON/10GEPON) standard with the multipoint control protocol (MPCP) `REPORT` messages [20]. In the opposite direction, the controller sends the schedule to all OIs by means of broadcast `Grant` messages, similar to the gate process with the MPCP `GATE` messages in EPON/10GEPON standard. The `Grant` and `Request` messages are therefore defined based on the frame structure of the MPCP `GATE` and `REPORT` messages, respectively [17]. The schedule duration T_{sc} consists of the running time of the ORS algorithm, the reconfiguration time of the controller, and the signaling time, i.e., time to transmit a `Grant` and respective `Request`s. It is important to note that in the case of a generic crossbar, the system needs to be reconfigured $N-1$ times in a working cycle resulting in the reconfiguration time of $(N-1)\alpha$, where α denotes the

reconfiguration delay (see Table I). Given that the distance between the OIs and controller in a DC is usually short, the propagation delay between them is assumed negligible. The computation of T_{sc} will be further detailed in Section IV. A.

C. Energy Saving Possibility

In the CORM, optical interfaces are not always busy with data transmissions and reconfigurations. Their idle time can be exploited for improving the overall energy efficiency with sleep mode operation. The idle time is mainly determined by the configurable accumulation period T_{ac}. In addition, an interface with light traffic load or no traffic may become idle during data transmission. The lower part of Fig. 2 shows the OI power model corresponding to the operation of the system in the upper part. Note that in the CORM the energy efficiency is improved by exploiting the idle time in a cycle, without affecting the network performance. The details about how the OI specifies its sleep time are provided in Section III-B2.

III. Optical Resource Scheduling Problem

Under the CORM framework, different ORS algorithms can be employed depending on the considered application scenario. This section studies in depth the optical resource scheduling problem for optical interconnection networks, e.g., in DCs. The problem is formulated as an open-shop scheduling problem. Existing non-preemptive and preemptive open-shop approaches are reviewed to highlight the need for the adaptive scheduling algorithm proposed in Section III-B.

A. Mapping ORS to Open-Shop Scheduling Problem

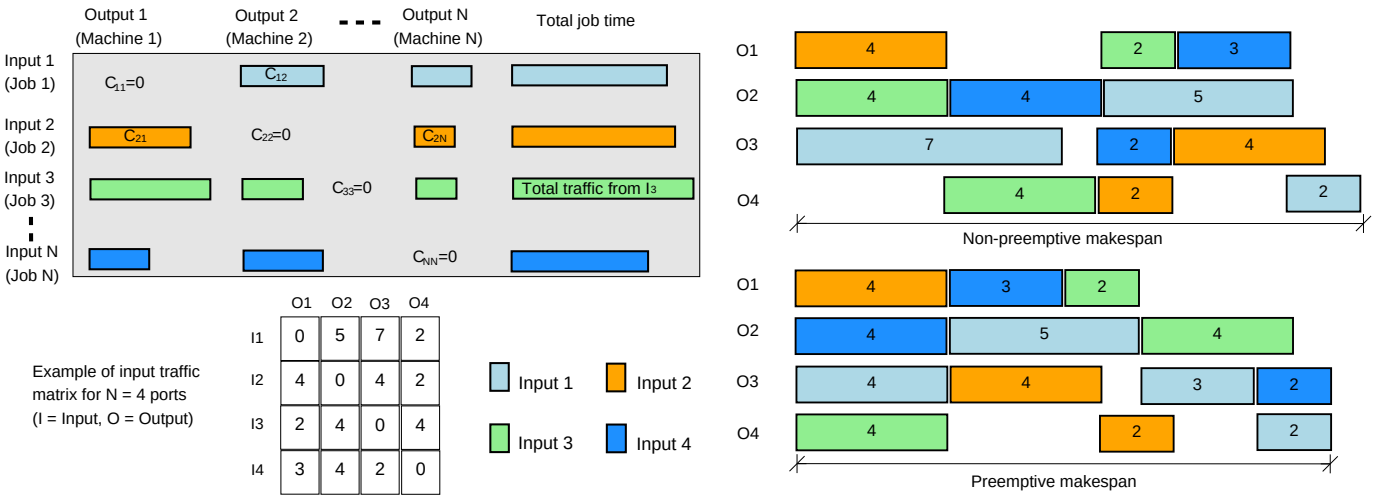
The considered ORS problem can be mapped to the classical open-shop scheduling problem [8], where the number of input ports is equal to that of output ports, as illustrated in Fig. 3(a). The input of an open-shop problem consists of a set of N jobs, a set of m machines ($m = N$), and a two-dimensional table describing the amount of time (greater or equal to zero) that each job should spend at a machine. Each job can be processed at only one machine at a time, and each machine can process only one job at a time. The jobs are processed in an arbitrary order. The goal is to assign jobs to machines, so that no more than one job is assigned to a machine at the same time, no job is assigned to more than one machine at the same time, and every job is assigned to each machine for the desired amount of time. The objective is to minimize the makespan, which is defined as the time from the start of the schedule to its end (i.e., the finishing time of the last job). The open-shop scheduling finds its applications in different contexts such as in inspection industry and health-care system [8], [21] or in optical networking [14]–[16]. For example, it is used to develop solutions for joint transmission scheduling and wavelength assignment in multichannel EPON [16] or scheduling transmissions in WDM networks with tunability [15].

To clarify the problem matching, input ports (transmitters) are considered as jobs and output ports (receivers) are considered as the machines. For an input traffic matrix C (i.e., the two-dimensional table) and identical reconfiguration delay α, the processing time of job_i on machine_j (i.e., task_{ij}) in the corresponding open-shop is $t_{ij} = \alpha + C_{ij}$ in case tasks are not dividable (i.e., non-preemptive scheduling). Since any valid open-shop schedule runs the task for t_{ij} time units, the input port i has sufficient time (α time units) to be reconfigured to connect to output port j and perform data transmission (C_{ij} time units). In the case of intra-rack communications, by assigning a dedicated wavelength to each optical receiver and the transmitters are scheduled to be tuned to different wavelengths to transmit traffic to the respective receivers, the intra-rack ORS problem can also be viewed as the open-shop problem [17]. Note, however, that when the number of wavelengths used in the system is less than the number of OIs, the mapping is no longer valid and thus another approach is needed to deal with the intra-rack scheduling problem.

The ORS problem can be solved using either existing preemptive or non-preemptive open-shop scheduling algorithms. As exemplified in Fig. 3(b), for a given input traffic matrix in Fig. 3(a) (lower part), the preemption enables jobs to be split providing more flexibility and thus a better scheduling performance (i.e., shorter makespan) than the non-preemptive algorithm. However, since jobs are divided and scheduled in noncontinuous time periods, preemptive algorithms incur extra reconfiguration overhead, which is significant when the delay is non-negligible. The larger the number of preemptions, the larger the overhead time. The overhead depends heavily on the reconfiguration delay, i.e., tuning time of the optical transceiver in the intra-rack and the switching time of the optical core switch in the inter-rack system, which is a hardware-dependent parameter.

An optimal schedule is the one which has the least finishing time among all possible schedules for a given input traffic matrix. It is proved in [22] that there exist polynomial time algorithms to obtain the optimal preemptive schedule for open-shops with negligible reconfiguration delay. Meanwhile, the non-preemptive open-shop problem with more than two machines is proven to be NP-complete [23]. Polynomial algorithms for optimal preemptive scheduling were introduced in [22], which are based on concepts from the theory of maximal matchings in bipartite graphs [24] and have the time complexity of $O(r^2)$, where r is the number of nonzero tasks (i.e., $r = N^2$ in the considered ORS problem). The optimal makespan is known to be the maximum line sum (both column and row), i.e., the largest value between the job lengths and machine processing times. On the other hand, there exist a number of non-preemptive greedy approximation algorithms. Among those, a well-known algorithm is the list-scheduling algorithm [8], which has an approximation ratio of about 2 (i.e., the makespan is about 2 times worst than the theoretical optimal value obtained for the preemptive case). In particular, a list-scheduling algorithm sorts a set of tasks in a given order and assigns a task in the unassigned list to the first machine that will become idle. A simple and often-used list-scheduling algorithm is the Longest Processing Time (LPT) algorithm, which sorts the jobs by their processing time and assigns an unassigned job to a machine with the earliest end time. This algorithm achieves an approximation ratio of $\frac{4}{3} - \frac{1}{3m}$, where m is the number of machines [25]. It

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2017.2695078, Journal of Lightwave Technology



(a) Illustration of open-shop problem matching (upper part) and an example of input traffic matrix (lower part)

(b) Illustration of preemptive vs. non-preemptive open-shop scheduling (negligible reconfiguration delay) for the input matrix on the left

Fig. 3. Mapping optical resource scheduling to open-shop problem.

is worth noting that while open-shop scheduling algorithms set the minimization of the makespan as the objective, there exist other scheduling approaches in the literature. For example, the wel-known iSLIP algorithm [18] is a typical iterative scheduling algorithm that trades scheduling performance with implementation simplicity and computation complexity.

The interconnection network architecture considered in this work can be viewed as a strictly non-blocking crossbar, where concurrent connections do not prevent connecting any available input ports to available output ports. However, it can be expected that the scheduling problem would become more complicated, when the system is reconfigurable non-blocking or with some level of blockingness. In such a scenario, further investigation regarding whether or not the open-shop mapping would still be valid is required.

B. Proposed Adaptive Open-Shop Scheduling Algorithm

This section first identifies the key factors that determine the comparison between the two scheduling strategies in terms of packet delay for given traffic condition and system configurations. Based on that analysis, an **A**daptive **O**pen-**S**hop scheduling algorithm (AOS) is then proposed that aims to take best of both preemptive and non-preemptive strategies in a dynamic and flexible fashion to maximize the scheduling performance for any given input traffic matrix and system parameters.

1) Determining Factors: In the CORM, given an input traffic matrix and a predefined accumulation period T_{ac}, the longer the makespan of the scheduling algorithm, the larger the packet delay. The makespan is therefore used to compare the two scheduling strategies in terms of packet delay performance. Let OPT denote the theoretical optimality obtained for the preemptive case (i.e., with negligible reconfiguration delay). OPT is identified as the maximum value of a so-called line sum, which is the sum of all entries in a row or a column [8]. N_{pre} denotes the number of preemptions of each OI in the optimal preemptive solution. Ω denotes the approximation ratio of the employed non-preemptive algorithm. The makespan achieved by the employed non-preemptive algorithm with negligible reconfiguration delay is referred to as ΩOPT. In addition, α denotes the reconfiguration time. The makespan of the preemptive and non-preemptive ORS algorithm is denoted by T_{ms}^{pre} and T_{ms}^{nopre}, respectively. It is assumed that OIs do not send traffic to themselves and all $N-1$ traffic demands of an OI in the input matrix are nonzero, i.e., each OI needs to reconfigure $N-1$ times in the case of non-preemptive scheduling strategy. Therefore, T_{ms}^{pre} and T_{ms}^{nopre} can be expressed as follows:

$$T_{ms}^{pre} = OPT + \alpha N_{pre}, \tag{1}$$

$$T_{ms}^{nopre} = \Omega OPT + \alpha(N-1). \tag{2}$$

From Eqs. (1) and (2), for a given input traffic matrix, an ORS algorithm should employ the non-preemptive approximation strategy when the number of reconfigurations N_{pre} is greater than N_0, which is a threshold derived by solving the $T_{ms}^{pre} = T_{ms}^{nopre}$ equation:

$$N_0 = \frac{(\Omega - 1)OPT}{\alpha} + N - 1. \tag{3}$$

Otherwise, the optimal preemptive scheduling strategy is employed for the input matrix. Therefore, the number of reconfiguration times of the adopted preemptive algorithm and N_0 are key determining factors in the ORS problem. From Eq. (3), as a reference value for comparing N_{pre} with, N_0 depends on several factors including not only the approximation ratio of the greedy algorithm Ω, but also the reconfiguration delay α, the input traffic matrix (i.e., OPT), as well as the number of OIs in the considered system N. Note that the value of OPT computed for an input traffic matrix depends also on the accumulation period T_{ac} and traffic profile. When the list-scheduling algorithm ($\Omega \approx 2$) is adopted for the non-preemptive scheduling, it is more beneficial to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2017.2695078, Journal of Lightwave Technology

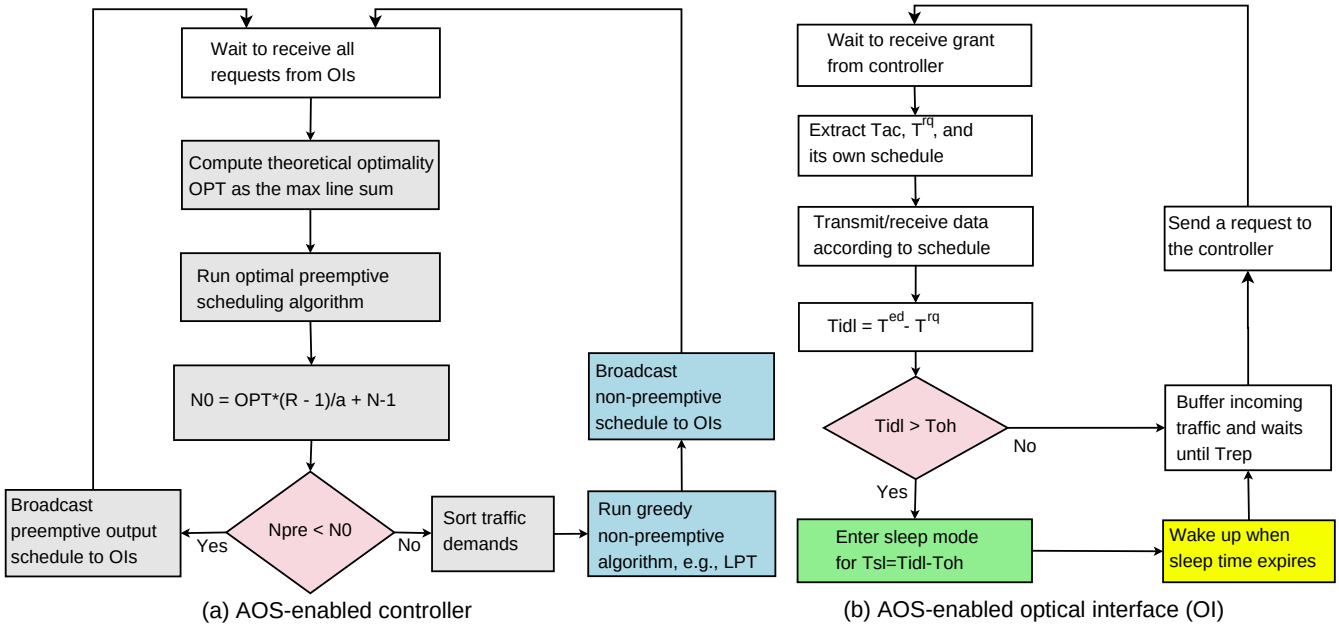


Fig. 4. Flowchart of proposed AOS algorithm (R and a are Ω and α in Eq. (3), respectively): (a) AOS-enabled controller; (b) AOS-enabled OI.

perform the preemptions when N_{pre} is less than or equal to $OPT/\alpha + N - 1$.

2) AOS Algorithm at Controller and Optical Interfaces: The main idea of the AOS algorithm is to always obtain the better performance by adaptively employing preemptive and non-preemptive scheduling strategies. More specifically, it adopts the preemptive schedule when the reconfiguration delay incurred from preemptions is lower than its performance improvement over the non-preemptive strategy and vice versa. That is, its expected makespan should be $T_{ms}^{aos} = \min\{T_{ms}^{pre}, T_{ms}^{nopre}\}$. Fig. 4 shows the flowcharts of the proposed AOS algorithm executed at the centralized controller and the OIs, both of which are further detailed in the following.

The flowchart of an AOS-enabled controller is shown in Fig. 4(a). The controller first waits and receives all request messages from the optical interfaces. The traffic demands contained in the requests help create the input traffic matrix, in which the theoretical optimality OPT can be identified as the maximum line sum (see Fig. 4(a)). In the next step, the controller runs the optimal preemptive scheduling algorithm, presented in [8], [22]. The resultant schedule is temporarily stored in the controller's memory. The scheduling threshold N_0 is then computed by using Eq. (3). In case the actual number of reconfigurations of the preemptive algorithm is lower than N_0, i.e., non-preemptive strategy is selected, the controller discards the preemptive schedule and further runs the adopted non-preemptive approximation algorithm, i.e., the LPT [25]. Thus, a list of traffic demands from the input matrix is first sorted in the descending order followed by the LPT execution, as detailed earlier. In both cases, the controller then broadcasts the output schedule to the OIs. It is important to note that the proposed AOS algorithm has the same polynomial computation complexity as the optimal preemptive algorithm, i.e., $O(N^4)$ since the greedy non-preemptive LPT algorithm has lower complexity, i.e., $O(N^2)$ [26].

The flowchart of an AOS-enabled OI is shown in Fig. 4(b). Basically, the OI operates according to the schedule of the AOS-enabled controller and determines operation mode itself according to its idle time and wake-up capability. As shown in Fig. 4(b), once the OI receives the grant message from the controller, it extracts its own schedule and the time instant it needs to send a request, computes its idle time based on the start of the next scheduled transmission, and justifies the possibility to enter sleep mode for improving its energy efficiency. To compute the sleep time in a cycle, let T^{ed} and T^{rq} denote the time instant when the OI completes all its data transmissions and the time instant when it starts sending a request to the controller for the next cycle, respectively. T^{ed} and T^{rq} are computed by the controller once the ORS algorithm is performed. In a cycle, each OI is idle for a duration $T_{idl} = T^{rq} - T^{ed}$ time units. If T_{idl} is longer than the time an OI requires to wake up from sleep period T_{oh} (i.e., wake up overhead time), the OI enters sleep mode for a duration $T_{sl} = T_{idl} - T_{oh}$ to save energy. It then must wake up in time, i.e., T_{oh} time units before the requesting time T^{rq} to send a new `Request` to the controller for the next cycle. Otherwise, the OI stays idle without sleeping. In both cases, incoming traffic is buffered during the idle time of the OI.

IV. Performance Analysis

This section presents the performance analysis of the proposed AOS algorithm. Performance metrics include average packet delay $\overline{D}$, i.e., the time from the moment a packet arrives at a sending OI until it is sent by the OI and energy saving η, the relative energy consumption decrease obtained from implementing sleep mode during idle time of the OIs. It is important noting that most of the existing studies on open-shop scheduling consider to minimize makespan as the objective. It is therefore desirable to translate the makespan obtained by a given scheduling algorithm to the average packet delay, which is a metric of interest in most communication networks.

A. Delay Analysis

In the CORM, the average packet delay $\overline{D}$ can be derived based on a graphical illustration (see Fig. 2). Without loss of generality, the packet delay of OI1 is considered. As shown in Fig. 2, a data frame arriving at the OI needs to experience several delay components. First, it needs to wait until being reported to the controller for a duration D_1. D_1 can be at most T_c and can be zero in case the arrival is immediately before the request message. On average, D_1 is $T_c/2$. Then, the packet experiences a delay from the request until the broadcast grant, called schedule time $D_2 = T_{sc}$. As mentioned earlier, this component depends on the reconfiguration time of the controller, the signaling time, and the runtime of the ORS algorithm. Since both scheduling approaches have polynomial computation complexity [22] and ORS runtime is assumed negligible, T_{sc} is approximated $T_{msg} + (N-1)\alpha$. Another delay component D_3 is the time from the moment the OI receives the grant until the packet (in red in Fig. 2) is transmitted. This duration could be zero, if the packet is scheduled to be transmitted first, but can be as much as the makespan T_{ms}, if it is the last packet in the schedule. As the scheduler aims at minimizing the makespan without following any policy to prioritize packets, assuming negligible packet time with respect to T_{ms}, the average value of D_3 is $T_{ms}/2$. Thus, by substituting T_c with $T_{ac} + T_{sc} + T_{ms}$, $\overline{D}$ can be expressed as:

$$\overline{D} = \frac{T_c + T_{ms} + 2T_{sc}}{2} = \frac{T_{ac} + 2T_{ms} + 3T_{sc}}{2}. \quad (4)$$

In the following, $\overline{D}$ is estimated for the two scheduling strategies and the AOS algorithm. Then, the energy-delay tradeoff is modeled to quantify the potential energy saving with sleep mode enabled for a given average delay constraint. For simplicity, uniform traffic distribution is considered.

1) Average Packet Delay in Preemptive Scheduling: In uniform traffic scenario, the optimality OPT becomes the time to transmit the amount of traffic of each OI in a scheduling cycle, which is equivalent to ρT_c, where $0 \leq \rho < 1$ represents the utilization factor of the system, i.e., the ratio of the total amount of traffic transmitted in time unit (also arrived traffic) over the total capacity of N channels. In steady state conditions, assuming negligible variance of OPT between two consecutive working cycles, it follows that:

$$\begin{aligned} OPT = \rho(T_{ac} + T_{ms} + T_{sc}) = \rho(T_{ac} + OPT + T_{sc}) \\ \Rightarrow\ OPT = \frac{\rho}{1-\rho}(T_{ac} + T_{sc}). \end{aligned} \quad (5)$$

From Eqs. (1), (4), and (5), the average delay in preemptive scenario $\overline{D}_{pre}$ is expressed as follows:

$$\overline{D}_{pre} = \frac{(1+\rho)T_{ac} + (3-\rho)T_{sc}}{2(1-\rho)} + \alpha N_{pre}. \quad (6)$$

The exact value of N_{pre} depends on actual input traffic matrix and is not easy to derive. It is therefore worth considering its upper and lower bounds. It is shown that any optimal algorithm for the multiprocessor scheduling problem uses at least 2(m-1) preemptions, where m is the number of machines [27]. This implies that the lower bound of N_{pre} is $2(N-1)$. On the other hand, since the open-shop problem can be considered as the switch scheduling problem with arbitrary reconfigurations [12], N_{pre} has another tighter upper bound, which is $N^2 - 2N + 2$. It follows that:

$$\begin{aligned} \frac{(1+\rho)T_{ac} + (3-\rho)T_{sc}}{2(1-\rho)} + 2\alpha(N-1) \leq \overline{D}_{pre} \\ \leq \frac{(1+\rho)T_{ac} + (3-\rho)T_{sc}}{2(1-\rho)} + \alpha(N^2 - 2N + 2). \end{aligned} \quad (7)$$

TABLE I
NOTATIONS, DESCRIPTIONS, AND DEFAULT VALUES

Notation	Description	Value and Unit
C	Channel capacity	10 Gb/s
N	Number of OIs in the system	16 – 48
T_{ac}	Accumulation period	8 – 2048 us
N_{pre}	Number of preemptions	variable
α	Reconfiguration delay	0.1 – 320 us
T_{msg}	Time to transmit a `Grant` and respective `Request` messages	1.024 us
ρ	Utilization factor (traffic load)	0.05 – 0.9
$\overline{D}_0$	Delay constraint	0.3 – 2 ms
T_{oh}	Wake-up overhead time of optical transceiver	20 – 200 us
P_a,P_{wk},P_s	Optical transceiver power consumption in active, wake-up, sleep states	5052 mW [28], 0.5P_a, 0.1P_a

2) Average Packet Delay in Non-preemptive Scheduling: In the case of non-preemptive scheduling algorithm, the average delay $\overline{D}_{nopre}$ is derived based on Eqs. (2), (4), and (5) as follows:

$$\begin{aligned} \overline{D}_{nopre} = \frac{(1-\rho+2\rho\Omega)T_{ac} + (3-3\rho+2\rho\Omega)T_{sc}}{2(1-\rho)} \\ +\alpha(N-1). \end{aligned} \quad (8)$$

Eventually, the average packet delay of the proposed AOS algorithm $\overline{D}_{aos}$ can be expressed as:

$$\overline{D}_{aos} = \min\left\{\overline{D}_{pre}, \overline{D}_{nonpre}\right\}, \quad (9)$$

where $\overline{D}_{pre}$ and $\overline{D}_{nonpre}$ are computed using Eqs. (6) and (8), respectively.

B. Energy-Delay Tradeoff

There exists an obvious trade-off between the packet delay and achievable energy saving in the CORM framework. T_{ac} is key to determine the trade-off. The longer the T_{ac}, the higher the energy saving, yet also the greater the packet delay. From Eq. (4), T_{ac} can be determined for a maximum allowable average packet delay $\overline{D}_0$. It follows that:

$$\overline{D} = \frac{T_{ac} + 2T_{ms} + 3T_{sc}}{2} \leq \overline{D}_0. \quad (10)$$

This equation is used to compute the maximum allowable value of T_{ac} for a given constraint $\overline{D}_0$ and a scheduling algorithm. Then, T_{ac} can be used to derive the maximum energy saving of the optical transceiver with sleep mode enabled. Energy saving η is computed within a cycle T_c. η is defined as the relative energy consumption decrease with respect to the energy consumption without sleep mode:

$$\eta = \frac{E - E'}{E}, \quad (11)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2017.2695078, Journal of Lightwave Technology

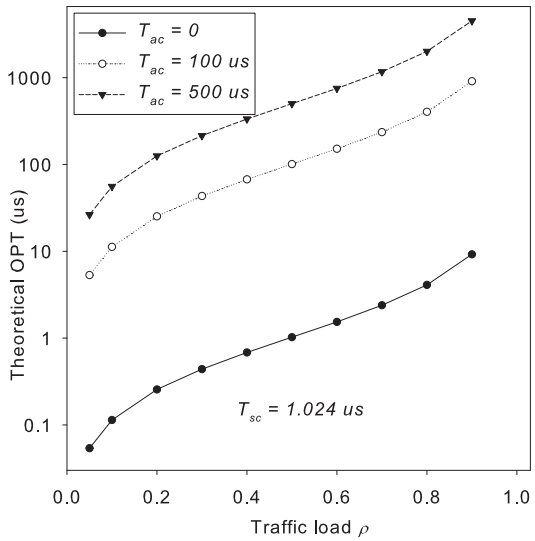


Fig. 5. Theoretical optimality as a function of traffic load ρ.

where E and E' denote energy consumption of an OI during a cycle, when the system is always active and when the sleep mode is enabled, respectively. Let P_a, P_s, and P_{wk} denote the OI power consumption in active, sleep, and wake-up states, respectively. E is expressed as:

$$E = T_c P_a. \tag{12}$$

During the wake-up process, OIs are assumed to consume half of the power in active mode (see Table I). For simplicity, it is assumed that the idle time T_{idle} is equal to T_{ac} and that T_{idle} is always greater than the wake-up overhead T_{oh}. E' is computed as follows:

$$E' = (T_{ac} - T_{oh})P_s + T_{oh}P_{wk} + (T_{sc} + T_{ms})P_a. \tag{13}$$

From Eqs. (12–13), Eq. (11) becomes:

$$\begin{aligned}\eta &= \frac{(T_c - T_{sc} - T_{ms})P_a - (T_{ac} - T_{oh})P_s - T_{oh}P_{wk}}{T_c P_a} \\ &= \frac{T_{ac}(P_a - P_s) - T_{oh}(P_{wk} - P_s)}{(T_{ac} + T_{sc} + T_{ms})P_a}.\end{aligned} \tag{14}$$

For preemptive scheduling, η^{pre} is computed with T_{ms}^{pre} derived from Eqs. (1) and (5), whereas T_{ac}^{pre} derived based on Eq. (10) and T_{ms}^{pre} as follows:

$$T_{ac}^{pre} = \frac{2(1-\rho)\overline{D}_0 - (3-\rho)T_{sc} - 2\alpha(1-\rho)N_{pre}}{1+\rho}. \tag{15}$$

For non-preemptive scheduling, η^{nopre} is computed with T_{ms}^{nonpre} derived from Eqs. (2) and (5), whereas T_{ac}^{nonpre} derived based on Eq. (10) and T_{ms}^{nonpre} as follows:

$$T_{ac}^{nopre} = \frac{(1-\rho)[2\overline{D}_0 - 2\alpha(N-1)] - (3-3\rho+2\rho\Omega)T_{sc}}{1-\rho+2\rho\Omega}. \tag{16}$$

Note that AOS is developed with the objective of delay minimization. However, when maximum energy saving is the objective with maximum allowable delay as the constraint (i.e., $\leq \overline{D}_0$), the AOS energy saving performance η^{aos} can be computed as the maximum between the η^{pre} and η^{nopre} .

V. Results

The performance evaluation is based on numerical analysis. The parameters and their default values are listed in Table I. The parameters are set in the context of intra- and inter-rack communications in DCNs [4], [6], [29]. LPT is employed as the non-preemptive ORS algorithm, i.e., $\Omega = (4N-1)/3N$ [25]. The theoretical optimality OPT is the key factor that determines the makespan T_{ms}^{aos} and, consequently, the delay performance. The dependence of OPT on the traffic load for different values of accumulation time is shown in Fig. 5. It can be seen that for the considered values of parameters, the theoretical makespan hardly reaches 1 ms. Actual delay and energy savings as function of different performance parameters are shown in the following.

Fig. 6(a) indicates the importance of an optimal preemptive algorithm with a few tuning/reconfigurations. In particular, with an example of intra-rack DC configuration (i.e., small values of α (0.2 us), N (48 OIs), and T_{ac} (0.2 ms)), a greater value of N_{pre} significantly increases the average preemptive packet delay. As a result, the AOS delay curve overlaps the preemptive one when N_{pre} is small, but it does not increase for larger N_{pre}. In addition, for a given value of N_{pre}, the higher the traffic load, the larger the delay. Fig. 6(b) shows the impact of reconfiguration delay α on the performance. As shown in the figure, AOS takes the preemptive scheduling approach when α is small. However, with large values of α, the AOS delay is equal to the non-preemptive one since the non-preemptive performs better than the preemptive. This is because the preemptive open-shop algorithm requires a significant number of preemptions resulting in high overall tuning overhead.

Fig. 7 shows delay performance for the case of inter-rack DC configurations as a function of the number of network interfaces N and reconfiguration delay α. Different from the small-scale scenario (e.g., intra-rack communication systems as shown in Fig. 6(a)), Fig. 7(a) shows that with considered set of parameters, for any number of OIs (i.e., network scale), the non-preemptive scheduling yields a lower packet delay for large-scale inter-rack DCNs. This is mainly because of high preemption overhead incurred in the preemptive approach when the number of involved network nodes is large (see also Eqs. (6-7)). Fig. 7(b) also confirms that the AOS takes non-preemptive scheduling strategy to have a lower delay when switching time of inter-rack devices are large, i.e., α is in order of 10 to 100 us. Together with Fig. 6(b), this highlights the importance of fast-reconfiguring devices, i.e., transceiver and switches in optical DCNs.

Fig. 8 shows the delay performance as a function of the traffic load and accumulation period for small-scale systems. Figs. 8 (a) and (b) collectively show that configuring a small value of accumulation period, the average packet delay is maintained below 1 ms for all the considered scenarios, an impressive delay performance in DC applications. The packet delay increases when traffic is heavier, as shown in Fig. 8(a), and/or a longer accumulation period is configured, as shown in Fig. 8(b). Note that the non-preemptive packet delay is compared with the upper and lower bounds of preemptive

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2017.2695078, Journal of Lightwave Technology

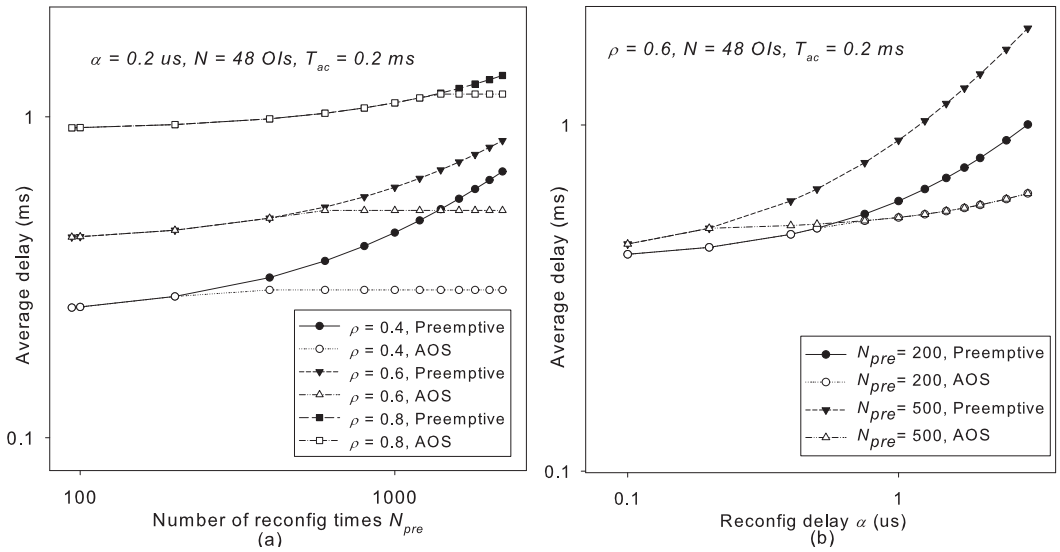


Fig. 6. Delay performance for intra-rack configurations: (a) Impact of reconfiguration times N_{pre} with various traffic loads; (b) Impact of reconfiguration delay α with various values of N_{pre}.

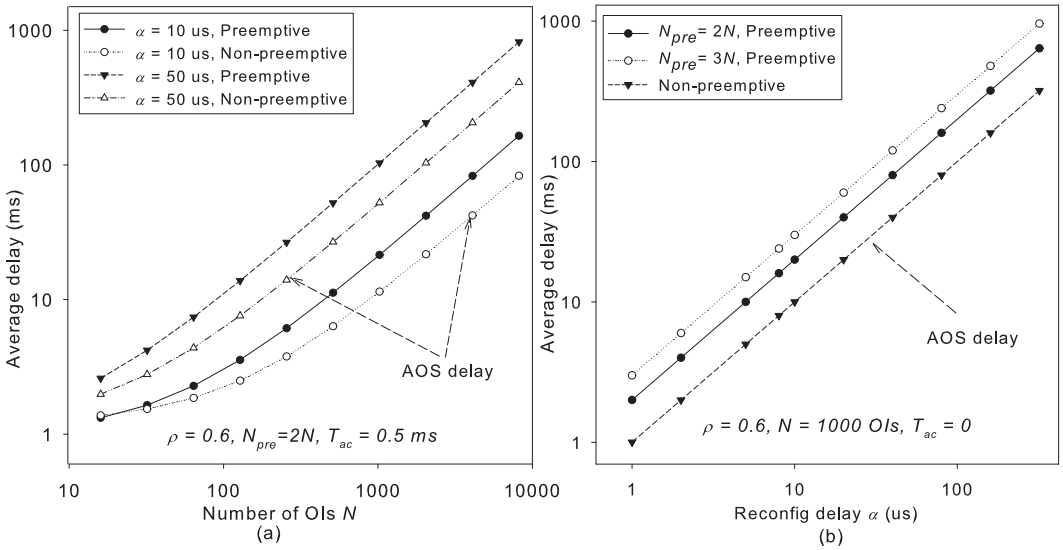


Fig. 7. Delay performance for inter-rack configurations: (a) Impact of OIs with $N_{pre} = 2N$; (b) Impact of α

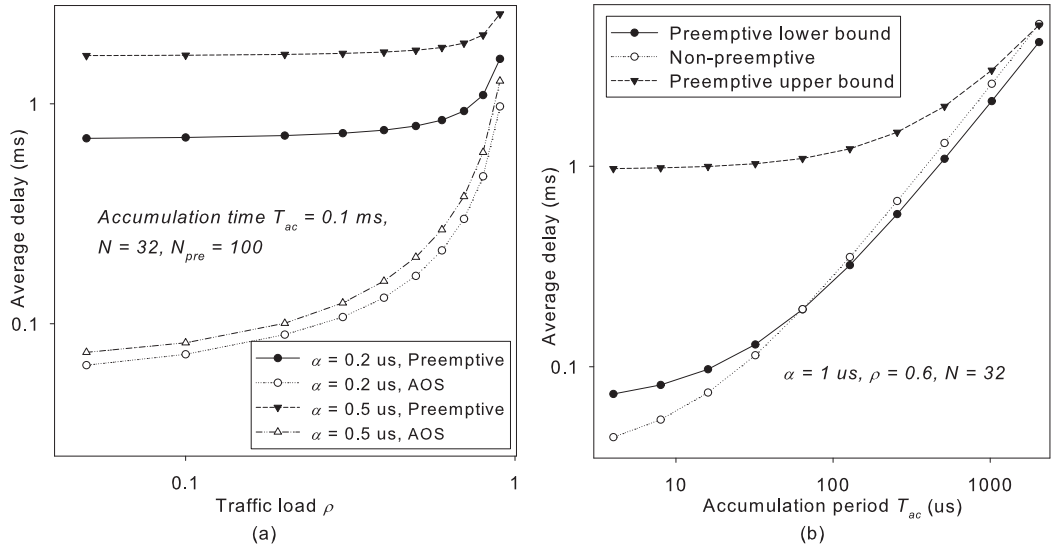


Fig. 8. Preemptive vs. AOS packet delay: (a) Impact of traffic load ρ; (b) Impact of accumulation period T_{ac}.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2017.2695078, Journal of Lightwave Technology

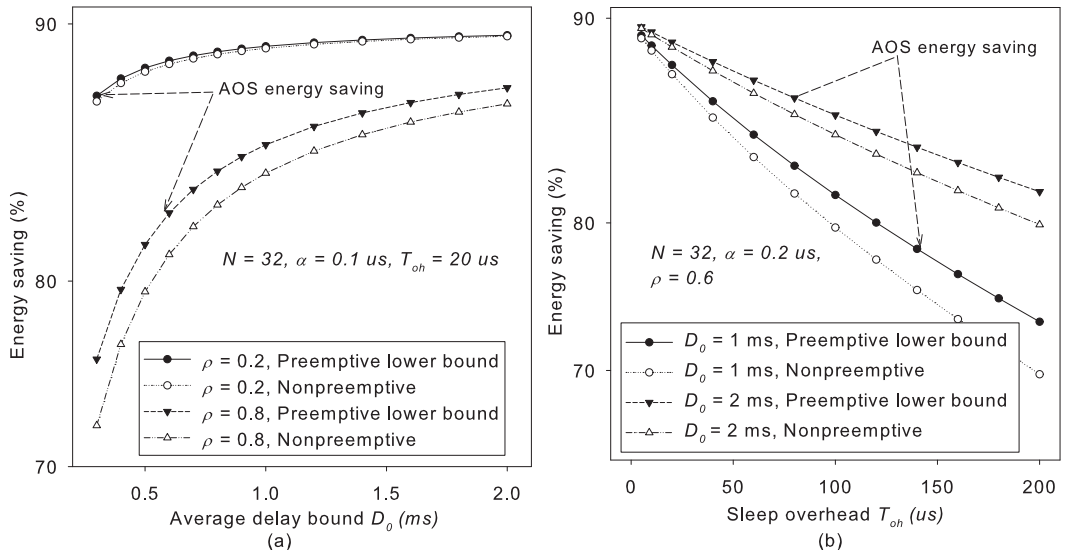


Fig. 9. Energy-delay tradeoff: (a) Energy saving vs. delay constraint for different traffic loads; (b) Impact of device wake-up capability T_{oh} on energy saving for various delay constraints.

packet delay in Fig. 8(b). It is observed that for considered values of parameters, non-preemptive scheduling outperforms the preemptive when T_{ac} is small. However, the opposite is true when the network is configured with a long T_{ac}. This further confirms that preemptive open-shop scheduling is recommended in applications whose delay requirements can be higher, e.g., inter-rack DC communications.

The tradeoff between achievable energy saving and packet delay is shown in Fig. 9. First, given the desirable short wake-up time of 20 us, significant amounts of energy can be saved even for strict delay requirements ($\overline{D}_0 \leq 2$ ms). The more relaxing the $\overline{D}_0$, the longer the T_{ac}, thereby more energy can be saved (see Fig. 9(a)). In addition, for a given value of $\overline{D}_0$, the lighter the traffic, higher energy saving can be obtained. Similar to the impact of the reconfiguration delay α on the delay performance, the wake-up capability of the optical transceiver T_{oh} has strong impact on the achievable energy saving, as shown in Fig. 9 (b). The longer the wake-up overhead time, the lower the energy saving. In particular, the energy saving decreases linearly with the increasing wake-up overhead. Figs. 9 (a) and (b) both show that with considered values of $\overline{D}_0$, the preemptive (lower bound) outperforms the non-preemptive in terms of energy savings.

VI. Conclusions

This paper studies the resource management problem in optical interconnection networks. A framework for flexible optical resource management has been proposed to facilitate the development and evaluation of efficient optical resource scheduling solutions. As a case study, the scheduling problem in inter- and intra-rack communications for DCs has been analyzed by applying the classical open-shop scheduling theory. The adaptive open-shop scheduling algorithm (AOS) has been proposed to dynamically select the optimal strategy between preemptive and non-preemptive scheduling, according to traffic condition and system parameters. An analytical model has been developed to quantify the delay performance and potential energy savings. The obtained results reveal that employing the existing greedy non-preemptive open-shop algorithms to schedule transmissions in optical DCNs is highly recommended, especially when the reconfiguration delay is long and where delay is a critical performance parameter, so minimization of delay needs to be prioritized. However, when energy saving is considered as the key objective under a given upper bound delay constraint, the preemptive scheduling approach outperforms the non-preemptive one. As a remark, even though the presented performance analysis and obtained results are carried out in the context of optical DCNs, they are directly applicable to any other open-shop problems with non-negligible reconfiguration time.

It is desirable to further reduce packet delay in optical interconnection networks. One possible enhancement is to improve the performance of optical hardware, i.e., reducing the switching time (or tuning time) of optical switches (or transceivers). For instance, semiconductor optical amplifier (SOA) based optical switches can provide switching time in the order of less than 10 ns [30]. SOA-based optical switches have a potential to achieve the similar latency performance as the electronic packet switches. New beam-steering optical switches for data centers with the switching time in the order of less than 150 us have also been demonstrated [31]. Meanwhile, optical transceivers with the tuning time below 200 ns have been experimentally demonstrated for data center applications [32]. Another possible option for improving latency performance is to reduce the control plane delay by using out-of-band signalling. This can be done by employing a centralized controller and can help to reduce the signalling overhead time. Furthermore, it would be interesting to compare the proposed AOS algorithm with other existing scheduling approaches in the literature such as the well-known iSLIP algorithm [18], which warrants future investigation.

Acknowledgment

This work was supported by the Swedish Foundation for Strategic Research (SSF), Swedish Research Council (VR), Göran Gustafssons Stiftelse, Natural Science Foundation of

Guangdong Province (Grant No. 508206351021), and National Natural Science Foundation of China (Grant No. 61550110240, 61671212).

REFERENCES

[1] K. Bergman and S. Rumley, "Optical switching performance metrics for scalable data centers," in *Proc., OECC/PS2016*, July 2016.

[2] Cisco White Paper, "Global cloud index: Forecast and methodology, 2014-2019."

[3] C. Kachris and I. Tomkos, "A survey on optical interconnects for data centers," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 1021–1036, Fourth 2012.

[4] K. Chen, A. Singla, A. Singh, K. Ramachandran, L. Xu, Y. Zhang, X. Wen, and Y. Chen, "OSA: An optical switching architecture for data center networks with unprecedented flexibility," *IEEE/ACM Trans. Netw.*, vol. 22, no. 2, pp. 498–511, Apr. 2014.

[5] R. Proietti, Z. Cao, C. J. Nitta, Y. Li, and S. J. B. Yoo, "A scalable, low-latency, high-throughput, optical interconnect architecture based on arrayed waveguide grating routers," *IEEE/OSA J. Lightw. Technol.*, vol. 33, no. 4, pp. 911–920, Feb. 2015.

[6] W. Ni, C. Huang, Y. L. Liu, W. Li, K. W. Leong, and J. Wu, "POXN: A new passive optical cross-connection network for low-cost power-efficient datacenters," *IEEE/OSA J. Lightw. Technol.*, vol. 32, no. 8, pp. 1482–1500, Apr. 2014.

[7] J. Chen, Y. Gong, M. Fiorani, and S. Aleksic, "Optical interconnects at the top of the rack for energy-efficient data centers," *IEEE Commun. Mag.*, vol. 53, no. 8, pp. 140–148, Aug. 2015.

[8] M. L. Pinedo, *Scheduling: Theory, algorithms, and systems*. Springer Science & Business Media, 2012.

[9] M. Fiorani, S. Aleksic, M. Casoni, L. Wosinska, and J. Chen, "Energy-efficient elastic optical interconnect architecture for data centers," *IEEE Commun. Lett.*, vol. 18, no. 9, pp. 1531–1534, Sept. 2014.

[10] T. Inukai, "An efficient SS/TDMA time slot assignment algorithm," *IEEE Trans. Commun.*, vol. 27, no. 10, pp. 1449–1455, Oct. 1979.

[11] B. Towles and W. Dally, "Guaranteed scheduling for switches with configuration overhead," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 835–847, Oct. 2003.

[12] X. Li and M. Hamdi, "On scheduling optical packet switches with reconfiguration delay," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 7, pp. 1156–1164, Sept. 2003.

[13] B. Wu, K. Yeung, P. han Ho, and X. Jiang, "Minimum delay scheduling for performance guaranteed switches with optical fabrics," *IEEE/OSA J. Lightw. Technol.*, vol. 27, no. 16, pp. 3453–3465, Aug. 2009.

[14] E. Bampis and G. Rouskas, "The scheduling and wavelength assignment problem in optical WDM networks," *IEEE/OSA J. Lightw. Technol.*, vol. 20, no. 5, pp. 782–789, May 2002.

[15] G. Rouskas and V. Sivaraman, "Packet scheduling in broadcast WDM networks with arbitrary transceiver tuning latencies," *IEEE/ACM Trans. Netw.*, vol. 5, no. 3, pp. 359–370, June 1997.

[16] L. Meng, J. El-Najjar, H. Alazemi, and C. Assi, "A joint transmission grant scheduling and wavelength assignment in multichannel SG-EPON," *IEEE/OSA J. Lightw. Technol.*, vol. 27, no. 21, pp. 4781–1492, Nov. 2009.

[17] D. Pham Van, M. Fiorani, L. Wosinska, and J. Chen, "Resource management for optical interconnects in data centre networks," in *Proc., GLOBECOM*, Dec. 2016.

[18] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches," *IEEE/ACM Trans. Netw.*, vol. 7, no. 2, pp. 188–201, Apr. 1999.

[19] J. Perry, A. Ousterhout, H. Balakrishnan, D. Shah, and H. Fugal, "Fastpass: A centralized "Zero-queue" datacenter network," in *Proc., SIGCOMM*, 2014, pp. 307–318.

[20] G. Kramer and G. Pesavento, "Ethernet passive optical network (EPON): building a next-generation optical access network," *IEEE Commun. Mag.*, vol. 40, no. 2, pp. 66–73, Feb. 2002.

[21] D. Bai, Z.-H. Zhang, and Q. Zhang, "Flexible open shop scheduling problem to minimize makespan," *Computers & Operations Research*, vol. 67, pp. 207–215, 2016.

[22] T. Gonzalez and S. Sahni, "Open shop scheduling to minimize finish time," *J. ACM*, vol. 23, no. 4, pp. 665–679, 1976.

[23] M. R. Garey, D. S. Johnson, and R. Sethi, "The complexity of flowshop and jobshop scheduling," *Math. Oper. Res.*, vol. 1, no. 2, pp. 117–129, May 1976.

[24] J. E. Hopcroft and R. M. Karp, "An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs," *SIAM J. Comput.*, vol. 2, no. 4, pp. 225–231, 1973.

[25] T. Gonzalez, O. H. Ibarra, and S. Sahni, "Bounds for LPT schedules on uniform processors," *SIAM J. Comput.*, vol. 6, no. 1, pp. 155–166, 1977.

[26] R. L. Graham, "Bounds on multiprocessing timing anomalies," *SIAM J. Appl. Math.*, vol. 17, no. 2, pp. 416–429, 1969.

[27] T. Gonzalez and S. Sahni, "Preemptive scheduling of uniform processor systems," *J. ACM*, vol. 25, no. 1, pp. 92–101, 1978.

[28] D. Pham Van, L. Valcarenghi, M. P. I. Dias, K. Kondepu, P. Castoldi, and E. Wong, "Energy-saving framework for passive optical networks with ONU sleep/doze mode," *Opt. Express*, vol. 23, no. 3, pp. A1–A14, Feb. 2015.

[29] H. Liu, F. Lu, A. Forencich, R. Kapoor, M. Tewari, G. M. Voelker, G. Papen, A. C. Snoeren, and G. Porter, "Circuit switching under the radar with REACToR," in *Proc., USENIX NSDI*, 2014, pp. 1–15.

[30] K. Ishii and S. Namiki, "Toward exa-scale photonic switch system for the future datacenter (invited paper)," in *Proc., IEEE/ACM NOCS*, Aug. 2016, pp. 1–5.

[31] W. M. Mellette, G. M. Schuster, G. Porter, G. Papen, and J. E. Ford, "A scalable, partially configurable optical switch for data center networks," *IEEE/OSA J. Lightw. Technol.*, vol. 35, no. 2, pp. 136–144, Jan. 2017.

[32] A. Funnell, J. Benjamin, H. Ballani, P. Costa, P. Watts, and B. C. Thomsen, "High port count hybrid wavelength switched tdma (ws-tdma) optical switch for data centers," in *Proc., OFC*, Mar. 2016, pp. 1–3.