Switch cost and packet delay tradeoff in data center networks with switch reconfiguration overhead

Shu Fu [a,d], Bin Wu [a,*], Xiaohong Jiang [b], Achille Pattavina [c], Hong Wen [d], Hongfang Yu [d]

[a] *School of Computer Science and Technology, Tianjin University, Tianjin 300072, PR China*
[b] *School of Systems Information Science, Future University Hakodate, Hakodate, Japan*
[c] *Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy*
[d] *National Key Laboratory on Communications, School of Communication and Information Engineering, University of Electronic Science and Technology of China, Chengdu 611731, PR China*

Received 1 June 2014
Revised 29 April 2015
Accepted 18 May 2015
Available online 28 May 2015

1. Introduction

With the fast development of broadband communications, modern network is shifting to support service-oriented applications rather than simply accommodating transmission connections. Most of those new applications are provided by data center networks (DCNs) [1–3], which consist of tens to thousands of clustered servers working in parallel to offer huge computing and storage resources. To provide the DCN service, data is generally processed in a distributed manner at different servers, and shared among them via a switch network.

Due to the huge number of servers in a DCN, scalable multi-stage switch networks such as Clos [4–8] or Fat-Tree [9–11] are adopted to achieve data switching among the servers. Fig. 1 shows a three-stage Clos network denoted by Clos(n, m, r). It has r input switches ($n \times m$), m middle switches ($r \times r$) and r output switches ($m \times n$). Switch ports

* Corresponding author. Tel.: +86 2227407133.
E-mail addresses: fshfshu@aliyun.com (S. Fu), binwu.tju@gmail.com (B. Wu), jiang@fun.ac.jp (X. Jiang), pattavina@elet.polimi.it (A. Pattavina), sunlike@uestc.edu.cn (H. Wen), yuhf2004@gmail.com (H. Yu).

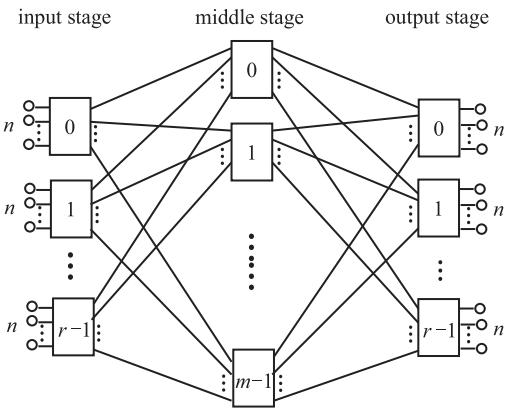


Fig. 1. Clos(n, m, r) network.

are connected by directed links, where the kth switch in a particular stage has a link connected to the kth input port of each switch in the next stage. By folding the network at the middle stage to merge the input and output stages into one, a Fat-Tree (a.k.a. folded-Clos) [9–11] can be obtained as a variation. Clos network and its variations are widely used for large-scale and system-level commodity interconnects due to their scalability nature, because the total number of required interconnects increases only moderately with n and r.

In a typical DCN [1–3], servers are organized into racks. A ToR (Top-of-Rack) switch [3] is equipped at each rack as the aggregation switch to collect/distribute data from/to the servers in the rack. In Fig. 1, the kth pair of input and output switches can be combined into a ToR switch at the kth rack, which can be routers or OpenFlow switches [12–14] capable of local in-rack data switching. For clarity, we still treat a ToR switch as a pair of separate input and output switches for ease of presentation. On the other hand, cross-rack traffic is handled by the middle switches which are generally crossbars.

More specifically, packet forwarding from a server (or a virtual machine inside) to that in another rack is based on some addressing techniques (such as source/destination addresses and port numbers). Packets are first collected by the ToR switch at the sending rack, and then sent to one of the middle switches which forwards the packets to the receiving rack ToR switch under the control of some traffic scheduling algorithms. Traffic scheduling at the middle switches can be carried out either in a coordinated manner using a centralized scheduling algorithm, or in a distributed manner where the scheduling processes of individual middle switches are independent of each other. In this work, we assume a centralized scheduler in DCNs and consider coordinated traffic scheduling across all middle switches. Though distributed traffic scheduling is an important and interesting research topic, it is left for future study.

Existing works on Clos network focus on finding proper system parameters and routing protocols to achieve cost-effective non-blocking communication, which ensures conflict-free connection (for circuit switching) or bandwidth utilization (for packet switching) for an arbitrary port pair between the input and output stages. In the circuit switching scenario, non-blocking conditions for strictly, wide-sense and rearrangeably non-blocking [6–8] have been studied. It is shown that the required number of middle switches is proportional to the number of ports of the aggregation switches in a fixed manner (e.g. $m \geq n$ for rearrangeably non-blocking). Packet switching based Clos network also inherits similar conditions to achieve conflict-free bandwidth utilization.

In Clos based DCNs, the above conditions impose a tight constraint on the number of middle switches and limit the flexibility of the switch network. In fact, a DCN may con-sist of hundreds of racks, each with tens to thousands of servers. Traffic amount is huge across the racks, though lim-ited at individual servers. This requires a large number of middle switches with enough ports and high switching ca-pacity for each. To cut down the system cost and make the DCN more scalable, it is desirable to reduce the number of middle switches.

The progress of optical transmission and high-capacity optical switching technologies provides a viable solution to support future scalable DCNs. At the ToR, low-rate traffic of the in-rack servers can be multiplexed onto fibers for high-speed optical transmission to the middle switches (i.e., core switches in practical DCNs). Meanwhile, high-capacity optical switches can be used as cores to switch the high-speed optical signals. This greatly cuts down the number of mid-dle switches and ports while simplifying interconnects in the DCN, leading to very good scalability with much reduced cost and power consumption.

Nevertheless, there is a *reconfiguration overhead* time in optical switches for tunable optical component adjustment and system synchronization, during which switching is idle and no data can be transmitted across the switch. Though such a reconfiguration overhead is generally trivial and thus ignored in a single electronic switch, it also exists in DCNs if a multi-stage electronic switching network is adopted, and coordinated scheduling/switching is desired among those electronic switches at different stages. In this case, additional overhead time is necessary for synchronizing ToR and middle switches when the latter is reconfigured. Due to the reconfiguration overhead, packet delay is inevitable.

In this paper, we adopt the batch scheduling approach [18–22] to study the packet scheduling problem in Clos net-work with switch reconfiguration overhead. By accumulat-ing packets over a period of time at the input switches, a traffic matrix $\boldsymbol{B}$ is obtained as a *batch* to denote the aggregated traffic among all pairs of input and output switches. Our approach is to use a centralized scheduler to decompose $\boldsymbol{B}$ into a set of permutation matrices, each of which denotes a crossbar configuration, and is assigned to a middle switch for independent execution. With the existing state-of-the-art algorithms to decompose $\boldsymbol{B}$, our work ensures *performance guaranteed switching* without packet loss and with a bounded packet delay. By reconfiguring each middle switch to independently fulfill multiple configurations, parallel multi-path switching is achieved by the middle switches. An intuitive observation is that, a larger number of middle switches lead to a smaller packet delay due to more switching paths available to shorten the batch switching time, and vice versa. That is why a tradeoff exists and the number of middle switches can possibly be reduced at the cost of a larger packet delay bound.

Our focus is on the tradeoff between the packet delay and the number of required middle switches m as in Fig. 1, and minimizing the cost of the switch network based on the tradeoff, as well as formulating criteria for choosing a proper traffic matrix decomposition algorithm to achieve cost minimization. Indeed, m is critical in determining the cost of the switch network. Since m equals to the number of out-put ports of each ToR switch, a larger m leads to more out-put ports of a ToR switch with a larger cost, and compli-cates the interconnects in the DCN. Besides, middle switches could be expensive as well if they are high-capacity opti-cal switches. Our work shows that m can be reduced and it does not necessarily satisfy $m \geq n$ in the proposed architec-ture. This makes the switch network design in DCNs more flexible, because the conventional non-blocking conditions for circuit-switching can be relaxed in the studied packet-switching scenario (in terms of the number of required mid-dle switches m).

Traffic matrix decomposition has been widely studied in [15–21]. The classical *Birkhoff-von Neumann* algorithm [15–17] decomposes an $r \times r$ matrix into $r^2 - 2r + 2$ permutations. By taking the reconfiguration overhead into account, much less permutations are generated in [18–21]. In particular, the traffic matrix is decomposed into a fixed number of $2r$ permutations by DOUBLE algorithm [18], and exactly r per-mutations by QLEF (Quasi Largest Entry First) [20]. Besides, ADAPT and SRF (Scheduling Residue First) algorithms are proposed in [21] to decompose the traffic matrix into a vari-able number of permutations with much improved schedul-ing efficiency. Our work adopts ADAPT [21] and QLEF [20] as two typical representatives due to their superior perfor-mance over others. As far as we know, this is the first effort to apply matrix decomposition to analyze cost-delay tradeoff in DCNs with multi-stage switching and switch reconfigura-tion overhead.

The rest of the paper is organized as follows. Section 2 describes our system model and problems. Section 3 formulates the tradeoff and the cost minimization. Section 4 gives more insights on our theoretical results by numerical analysis. Implementation issues and discussions are presented in Section 5. We conclude the paper in Section 6.

2. System model and problems

2.1. System model

Our system model is based on the three-stage Clos(n, m, r) in Fig. 1 with OpenFlow switches [12–14] as the ToR switches. An OpenFlow ToR switch combines a pair of input and output switches. It is capable of collecting/distributing packets from/to the servers in the rack by proper port matching. Besides, the OpenFlow ToR switches are buffered, such that outgoing packets of the racks can be accumulated over time, and be presented onto the output ports of the ToR switches in a parallel manner to support simultaneous multi-path switching across the middle stage. Meanwhile, incoming packets of the racks can be buffered before being distributed to the servers. We consider a time slotted system and assume that each time slot can accommodate one packet.

The middle switches are crossbars for handling cross-rack traffic, each of which has a reconfiguration overhead of δ timeslots. The cross-connection status of a middle switch is defined as a *configuration* and is denoted by an $r \times r$ permutation matrix. Each row of the permutation matches an input port of the middle switch and each column an output port, whereas an entry of 1 means a connection of the two corresponding ports and 0 otherwise.

Besides, each link connecting a pair of ports between ToR and middle switches is a high-speed optical interconnect, which can transmit packets at a rate M times faster than the data rate of a server due to optical multiplexing. For example, in large-size DCNs with thousands of servers per rack, time-slotted WDM (wavelength division multiplexing) can serve as an efficient optical multiplexing technique. Without loss of generality, we assume $M = n$ in this work whereas our results can be easily extended to other values of $M > n$ as well.

The $r \times r$ traffic matrix $\boldsymbol{B}$ is obtained by periodically accumulating packets at the input switches over a period of A timeslots, resulting in at most nA outgoing packets at each input switch. A centralized scheduler is assumed to perform traffic matrix decomposition for generating a set of middle switch configurations, which are assigned to different middle switches for achieving parallel and independent multipath switching. To facilitate system modeling and theoretical analysis, we assume that the total number of packets destined to each output port of any output switch is no more than A, and that arrived at each output switch is at most nA. This assumption is similar to that in [18–21] for ensuring proper formulation of traffic matrix decomposition. Due to the bursty and statistical nature of the traffic, it can be ensured by multiple ways in engineering practice, such as choosing a proper value of A, or involving differentiated performance guarantee (i.e., different values of A for different flows), etc.

With the above definitions and assumptions, each row of $\boldsymbol{B}$ matches an input switch and each column an output switch. An entry in $\boldsymbol{B}$ denotes the number of packets to be transmitted between the corresponding switch pair. Accordingly, the maximum line (row or column) sum of $\boldsymbol{B}$ equals to nA. Note that those nA packets can be transmitted across the middle switches in A timeslots using the high-speed optical interconnect since $M = n$ is assumed in this work. This makes our model exactly the same as those in [18–21], where performance guaranteed switching can be achieved with a bounded packet delay of $3A+H$ (H is a constant time required to run the scheduling algorithm). The whole packet switch-ing process consists of four stages: traffic accumulation in A time slots, running the scheduling algorithm in H time slots, packet switching across the switch network in A time slots, and transmitting packets from output switch buffers to out-put lines in another A time slots. A timing diagram for the four stages can be found in [18–21]. In what follows, we slightly abuse A as the packet delay (bound) for simplicity.

2.2. Problems

In addition to formulating the tradeoff between packet delay and switch cost (denoted by the number of middle switches m), we also study how to choose a proper ma-trix decomposition technique (either ADAPT [21] or QLEF [20]) to minimize either m or an overall cost of the switch network under a given set of system parameters. To gauge the overall cost of the switch network, we define a cost

function between switch cost (denoted by m) and packet delay bound (denoted by A) to make the two cost factors comparable with each other, which allows us to combine them into an integrated metric for cost minimization. This is reasonable in practical DCNs, since packet delay matches QoS (Quality of Service) which could be related to either the DCN performance or the revenue of providing the DCN service. In brief, this paper addresses the following problems.

- Formulate the tradeoff between the packet delay bound and the number of middle switches m.
- Minimize the cost of the switch network denoted by m under a given packet delay bound.
- Minimize an overall cost with packet delay translated into a cost factor comparable with the switch cost.
- Derive criteria for choosing a matrix decomposition technique to generate middle switch configurations with the objective of cost minimization.

3. Tradeoff and cost minimization

The tradeoff between switch cost (denoted by m) and packet delay can be formulated under a specific matrix decomposition technique. As we have mentioned in Section 1, ADAPT [21] and QLEF [20] are considered in this work. In what follows, we first briefly summarize the theoretical as-pects of the two algorithms in Section 3.1. The detailed ma-trix decomposition processes are indeed not important at this stage of theoretical analysis, and thus will be discussed later in Section 5 (see Figs. 5 and 6). Based on theories in ADAPT and QLEF, we formulate the tradeoff and the cost minimization problems under ADAPT in Sections 3.2–4 and QLEF in Section 3.5. The criteria for choosing a proper matrix decomposition technique for cost minimization are studied in Section 3.6. Table I summarizes the notations used in this paper.

3.1. Summary of ADAPT and QLEF algorithms

If a matrix $\boldsymbol{B}$ is decomposed into a set of permutations $\{\boldsymbol{P_k}\}$ with a weight φ_k for each $\boldsymbol{P_k}$, and the sum of all $\varphi_k \boldsymbol{P_k}$ is not smaller than $\boldsymbol{B}$ at every entry, we say that $\boldsymbol{B}$ is *covered* by the set of permutations $\{\boldsymbol{P_k}\}$ with corresponding weights $\{\varphi_k\}$.

Given an $r \times r$ traffic matrix $\boldsymbol{B}$ with a maximum line sum of nA, ADAPT algorithm [21] can generate a set of N_S ($r^2 - 2r + 2 > N_S > r$) permutations $\{\boldsymbol{P_k}\}$ ($N_S \geq k \geq 1$) to cover $\boldsymbol{B}$, with each $\boldsymbol{P_k}$ equally weighted by

$$\varphi_k = \frac{nA}{N_S - r}. \tag{1}$$

Note that $N_S = r$ matches QLEF [20] and $N_S = r^2 - 2r + 2$ matches Birkhoff-von Neumann decomposition [15–17]. More specifically, ADAPT in [21] also determines the best N_S value for single switch scheduling. Since we consider multiple parallel middle switches in Fig. 1, we only adopt the matrix decomposition technique in ADAPT but determine the best N_S value using a different mechanism.

On the other hand, QLEF algorithm [20] generates exactly $N_S = r$ permutations $\{\boldsymbol{P_k}\}$ to cover $\boldsymbol{B}$ with a *worst-case* weight φ_k in (2) for each $\boldsymbol{P_k}$.

$$\begin{aligned}
&\varphi_{k+1}|_{\lceil \frac{r}{2} \rceil > k \geq 0} \\
&= \max\left\{ \left. \frac{nA}{\left\lceil \frac{k-\Delta}{2} + 1 \right\rceil} \right|_{\Delta = \frac{(3r-1) - \sqrt{(3r-1)^2 - 8(r-1)(k+2)}}{4}}, \left. \frac{nA}{\left\lceil \frac{k-\Delta}{2} + 1 \right\rceil} \right|_{\Delta = \frac{2k^2 + k + 2r}{2r+1}} \right\}, \\
&\varphi_{k+1}|_{r \geq k+1 \geq \lceil \frac{r}{2} \rceil} = \varphi_{\lceil \frac{r}{2} \rceil - 1}.
\end{aligned} \tag{2}$$

3.2. ADAPT based switch cost and packet delay tradeoff

Assume that ADAPT [21] is used to decompose the traffic matrix $\boldsymbol{B}$ into N_S permutations. Due to the reconfiguration overhead δ of the middle switches and the assumption of $M = n$ (i.e., the multiplexing factor), we have

$$\frac{1}{m}\left(\delta N_S + \frac{1}{M}\sum_{k=1}^{N_S} \varphi_k\right) = \frac{1}{m}\left(\delta N_S + \frac{1}{n}\sum_{k=1}^{N_S} \varphi_k\right) = A, \tag{3}$$

where m is the number of middle switches, and φ_k is the number of timeslots that a configuration $\boldsymbol{P_k}$ should be kept for packet transmission. A is the traffic accumulation time. Packet switching across the middle switches must be completed in A timeslots to ensure that the ToR buffers are not overwhelmed. Define *speedup* as the ratio of packet transmission rate inside a switch over the rate outside at the input port of the switch. With multi-path routing, it is ensured in (3) that no additional frequency domain speedup is required at each middle switch.

In (3), $\delta N_S + \frac{1}{M}\sum_{k=1}^{N_S} \varphi_k$ is called the *scheduling length*. In [18–21], it is the total number of timeslots required for transmitting all packets across a single switch. In our case, packet switching is carried out by m parallel middle switches, and thus the scheduling length is averaged over the m switches. This may lead to a *truncation error*. In other words, we may not be able to exactly and equally average the scheduling length over the m middle switches, while ensuring each configuration to be fulfilled only by a single switch. However, such a truncation error is mainly an engineering concern and it can only trivially bias our theoretical results (detailed in Section 5.4). Other than the truncation error that may occur in the averaging process as in (3), we do not allow a configuration to be separated and fulfilled by two middle switches.

Eqs. (1) and (3) lead to

$$A = \frac{\delta N_S}{m - \frac{N_S}{N_S - r}}; \tag{4}$$

and

$$m = \frac{\delta N_S}{A} + \frac{N_S}{N_S - r}. \tag{5}$$

It is clear in (4)–(5) that a larger A, which corresponds to a larger packet delay, matches a smaller number m of middle switches, and vice versa.

3.3. ADAPT based switch cost minimization for a given delay

In this part, we assume that ADAPT is adopted and the set of parameters $\{\delta, r, A\}$ are given, where A matches a packet delay bound. The objective is to find the minimum number of middle switches m to satisfy the delay bound.

Table. I
List of notations and their definitions.

Notations	Definitions
N	Number of input ports of an input stage ToR switch, or number of output ports of an output stage ToR switch.
M	Number of middle switches.
R	Number of ToR switches.
M	Multiplexing factor for an optical connection between a pair of ToR and middle switch ports.
B	Traffic matrix.
A	Traffic accumulation time.
H	Time for running the scheduling algorithm.
P_k	A middle switch configuration as indexed by k.
φ_k	Lasting time of configuration $\boldsymbol{P_k}$.
N_S	Number of configurations in the schedule generated.
δ	Reconfiguration overhead time for each configuration.
P	System parameter as defined in (10).
τ	Per-unit-delay cost.
C	Overall cost combining switch and delay costs.
S	A φ_k related factor defined in (17) for QLEF.
$A_{\min}^{\text{QLEF}}$	Minimum value of A for QLEF.
$A_{\min}^{\text{ADAPT}}$	Minimum value of A for ADAPT.
$\Delta A_{\min}$	$\Delta A_{\min} = A_{\min}^{\text{QLEF}} - A_{\min}^{\text{ADAPT}}$.
$m_{\min}^{\text{QLEF}}$	Minimum value of m for QLEF.
F	A ratio based metric defined in (35).
$f_{\min}$	Minimum value of f.
C_{ADAPT}	Overall cost C for ADAPT.
C_{QLEF}	Overall cost C for QLEF.
F	F=$C_{\text{ADAPT}}/C_{\text{QLEF}}$ as defined in (40).

By (5), we get the second order differential of m over N_S as

$$\frac{d^2m}{dN_S{}^2} = \frac{2r}{(N_S - r)^3} > 0. \tag{6}$$

The inequality in (6) holds because $N_S > r$ must be true in order to achieve performance guaranteed switching [18–21]. As a result, m in (5) is a concave function of N_S and we can find a unique value of N_S to minimize m.

Let

$$\frac{dm}{dN_S} = \frac{\delta}{A} - \frac{r}{(N_S - r)^2} = 0. \tag{7}$$

We get

$$N_S = r + \sqrt{\frac{rA}{\delta}}. \tag{8}$$

Note that N_S is intrinsically an integer but is treated as a real value in theoretical analysis. In practice, we can use $\lceil N_S \rceil$ to replace N_S obtained in (8).

By using ADAPT to decompose $\boldsymbol{B}$ into N_S permutations (see (8)), and fulfilling those permutations in parallel using the m middle switches, we can minimize m as

$$m = \frac{\delta N_S}{A} + \frac{N_S}{N_S - r}\bigg|_{N_S = r + \sqrt{\frac{rA}{\delta}}} = (1 + p)^2, \tag{9}$$

where

$$p = \sqrt{\frac{\delta r}{A}}. \tag{10}$$

As we have discussed in Section 1, the circuit switching based Clos(n, m, r) network requires $m \geq n$ for achieving rearrangeably non-blocking. Obviously, in our proposed architecture $m < n$ can be obtained based on (9) if $p < \sqrt{n} - 1$, which can be achieved by enlarging A as per (10).

3.4. ADAPT based overall cost minimization with delay cost

In Section 3.3, we have minimized m under a given packet delay A. In fact, DCN service providers may need a mechanism to determine a proper packet delay bound, as it is directly related to both the QoS and the switch cost (due to the tradeoff). As discussed in Section 2.2, we can define a cost function between switch cost (denoted by m) and packet delay bound (denoted by A) to combine the two cost factors into an integrated *overall cost* C. In practice, this integrated cost function may take various forms (e.g., linear and nonlinear) based on different engineering considerations of the DCN op-erators. For simplicity, we assume the linear cost function in (11) by defining a *per-unit-delay cost* τ to make the two cost factors comparable with each other.

$$C = m + \tau A. \tag{11}$$

Note that m is not an independent variable in (11). Instead, it is a function of A as formulated in (9)–(10). As a result, we have

$$C = m + \tau A = (1 + p)^2 + \tau A = \left(1 + \sqrt{\frac{\delta r}{A}}\right)^2 + \tau A. \tag{12}$$

At this point, our objective is to find a proper value of A to minimize the overall cost C in (11)–(12).

Similar to the analysis in Section 3.3, we have

$$\frac{d^2C}{dA^2} = \frac{1}{A^3}\left(\frac{3\sqrt{\delta rA}}{2} + 2\delta r\right) > 0. \tag{13}$$

So, there exists a unique value of A to minimize C, which is the solution of (14) and can be obtained by numerical searching.

$$\frac{dC}{dA} = -\frac{\delta r}{A^2} - \sqrt{\frac{\delta r}{A^3}} + \tau = -\frac{1}{\delta r}\left(p^4 + p^3 - \tau\delta r\right) = 0. \quad (14)$$

After p and A are obtained by solving (14), we can determine N_S in (8), m in (9) and C in (11). Then, ADAPT decomposition is used to generate N_S configurations to minimize the overall cost C.

3.5. QLEF decomposition based results

QLEF (Quasi largest entry first) algorithm is originally pro-posed in [20] to schedule traffic in a single switch with re-configuration overhead, where it minimizes the packet delay bound by using the minimum number of $N_S = r$ switch configurations. With multiple parallel middle switches in our case, the situation is quite different as discussed below.

Since QLEF ensures $N_S = r$, we assume that $m \le r$, and Eq. (3) translates to

$$\frac{1}{m}\left(\delta r + \frac{1}{n}\sum_{k=1}^{r}\varphi_k\right) = A, \quad (15)$$

where $\varphi_k (r \ge k \ge 1)$ can be calculated in (2) [20] based on the system parameters A and r. Eq. (15) is equivalent to

$$\frac{1}{m}\left(\frac{\delta r}{A} + \frac{1}{nA}\sum_{k=1}^{r}\varphi_k\right) = \frac{1}{m}\left(\frac{\delta r}{A} + S\right) = 1, \quad (16)$$

where

$$S = \frac{1}{nA}\sum_{k=1}^{r}\varphi_k \quad (17)$$

is a constant which only depends on r. Based on (16)–(17), the tradeoff between A and m can be formulated in (18)–(19), but subject to $m \le r$.

$$A = \frac{\delta r}{m - S}; \quad (18)$$

$$m = \frac{\delta r}{A} + S = p^2 + S. \quad (19)$$

Accordingly, the minimum achievable values of A and m are

$$A_{\min}^{\mathrm{QLEF}} = \frac{\delta r}{r - S}; \quad (20)$$

and

$$m_{\min}^{\mathrm{QLEF}} > S. \quad (21)$$

Meanwhile, the overall cost C in (11) translates to

$$C = m + \tau A = p^2 + S + \tau A = \frac{\delta r}{A} + S + \tau A, \quad (22)$$

and thus

$$\frac{d^2C}{dA^2} = \frac{2\delta r}{A^3} > 0. \quad (23)$$

Therefore, the value of A for minimizing C is the solution of

$$\frac{dC}{dA} = \tau - \frac{\delta r}{A^2} = 0, \quad (24)$$

which is

$$A = \sqrt{\frac{\delta r}{\tau}}. \quad (25)$$

From (19), (22) and (25), the minimum overall cost C and the achieving number of the middle switches m are

$$m = \frac{\delta r}{A} + S\Big|_{A=\sqrt{\frac{\delta r}{\tau}}} = \sqrt{\delta r\tau} + S; \quad (26)$$

and

$$C = m + \tau A|_{A=\sqrt{\frac{\delta r}{\tau}}} = 2\sqrt{\delta r\tau} + S. \quad (27)$$

Also note that τ must be properly constrained in order to use (25)–(27) to calculate A, m and C in the QLEF scenario. This is because $m \le r$ entails

$$m = \sqrt{\delta r\tau} + S \le r. \quad (28)$$

Therefore, the following (29) must be satisfied.

$$\tau \le \frac{(r - S)^2}{\delta r}. \quad (29)$$

Otherwise, A will take its boundary value as in (20) with $m = r$.

3.6. Operation zones of ADAPT and QLEF

Theorem 1. *ADAPT decomposition should be adopted if* $A_{\min}^{\mathrm{QLEF}} > A \ge A_{\min}^{\mathrm{ADAPT}}$ *where* $A_{\min}^{\mathrm{QLEF}}$ *is in* (20) *and* $A_{\min}^{\mathrm{ADAPT}} = 4\delta r/(\sqrt{1+4r}-1)^2$, *and QLEF must meet* $r > S$.

Proof. If $A < A_{\min}^{\mathrm{QLEF}}$, QLEF is infeasible by our assumption. Instead, ADAPT can be feasible according to (8)–(10), though m could be relatively large. On the other hand, the assumption of $r \ge m$ and (21) in QLEF lead to $r > S$. The remaining issue is to prove that the minimum achievable A in ADAPT is

$$A_{\min}^{\mathrm{ADAPT}} = \frac{4\delta r}{\left(\sqrt{1+4r}-1\right)^2}. \quad (30)$$

As A in ADAPT decreases, N_S decreases and m increases according to (8)–(10). Since we do not allow a configuration to be separated and fulfilled by two middle switches (other than in the averaging process of the scheduling length as in (3) and (15)), the minimum achievable A in ADAPT is determined by $m = N_S$(where each middle switch exactly fulfills a single configuration). Based on (8)–(10), we have

$$(1+p)^2 = r + \frac{r}{p}, \quad (31)$$

and

$$p = \sqrt{\frac{\delta r}{A}} = \frac{\sqrt{1+4r}-1}{2}. \quad (32)$$

Note that we have $A = A_{\min}^{\mathrm{ADAPT}}$ in (32) and thus (30) is obtained. □

Theorem 2. *If we only focus on minimizing m under a given A, QLEF decomposition should be adopted for* $4\delta r/(S-1)^2 > A \ge \max\{A_{\min}^{\mathrm{QLEF}}, A_{\min}^{\mathrm{ADAPT}}\}$ *with S in (17) and* φ_k *in (2). Otherwise, ADAPT should be adopted if* $A \ge 4\delta r/(S-1)^2$.

Proof. When $A \geq \max\{A_{\min}^{\text{QLEF}}, A_{\min}^{\text{ADAPT}}\}$, both ADAPT and QLEF are feasible. On the other hand, $4\delta r/(S-1)^2 > A$ entails

$$p = \sqrt{\frac{\delta r}{A}} > \frac{S-1}{2} \tag{33}$$

and thus

$$(1+p)^2 > p^2 + S \tag{34}$$

According to (9) and (19), this means that ADAPT requires a larger m than QLEF and thus QLEF should be adopted for traffic matrix decomposition. In contrast, ADAPT should be used for $A \geq 4\delta r/(S-1)^2$.

The above Theorems 1–2 depict the operation zones of ADAPT and QLEF under a given A, but they do not reveal the performance gap between the two. To characterize the gap, we define a metric f in (35) for the switch cost minimization scenario without considering the cost of packet delay.

$$f = \frac{p^2 + S}{(1+p)^2}. \tag{35}$$

Based on (9) and (19), f is the ratio of m in QLEF over that in ADAPT under a given delay A.

Theorem 3. *If we only focus on minimizing m under a given A,* Eq. (36) *holds in the zones where both ADAPT and QLEF are feasible.*

$$\lim_{p\to+\infty} f = 1 \quad \text{and} \quad \lim_{p\to 0} f = S. \tag{36}$$

Theorem 3 tells us that if $p \to +\infty$ (e.g., $r \to +\infty$ with given δ and A in (10)), m required by QLEF and ADAPT tends to be the same. If $p \to 0$ (e.g., $A \to +\infty$ with given δ and r in (10)), m in QLEF is S times larger than that in ADAPT. Note that S is solely determined by r according to (2) and (17).

Theorem 4. *For a given r, f in* (35) *is minimized as* $f_{\min} = p/(1+p) = S/(1+S)$ *at* $p = S$ *with S in* (17)*, which means that m in QLEF is at least* $S/(1+S)$ *times of that in ADAPT in the zones where both decompositions are feasible.*

Proof. Since S is solely determined by r according to (2) and (17), a given r means a constant S in (35). As a result, p becomes the only variable to determine f in (35). Let

$$\frac{df}{dp} = \frac{2(p-S)}{(1+p)^3} = 0, \tag{37}$$

we have $p = S$. Moreover, we have

$$\frac{d^2 f}{dp^2} = \frac{2 - 4p + 6S}{(1+p)^4}, \tag{38}$$

which is positive at $p = S$. Consequently, f is minimized as the following $f_{\min}$ at $p = S$.

$$f_{\min} = \left.\frac{p^2+S}{(1+p)^2}\right|_{p=S} = \frac{p}{1+p} = \frac{S}{1+S}. \tag{39}$$

□

Theorem 5. *Assume that the overall cost metric is adopted for choosing the matrix decomposition technique. Let A be the solution of* (14) *and C(A) be the corresponding ADAPT based overall cost as formulated in* (12)*. When* τ *satisfies* (29)*, ADAPT decomposition should be used if* $C(A) < 2\sqrt{\delta r \tau} + S$ *and QLEF should be used otherwise. When* (29) *is not satisfied, this condition changes to* $C(A) < r + \delta r \tau / (r - S)$.

Proof. When τ satisfies (29), the minimum overall cost C in QLEF can be calculated in (27) as $C = 2\sqrt{\delta r \tau} + S$. Otherwise, A in QLEF will be $A_{\min}^{\text{QLEF}}$ as in (20) with $m = r$, which leads to $C = m + \tau A = r + \delta r \tau / (r - S)$. In either case, ADAPT should be used if $C(A) < C$ and QLEF otherwise.

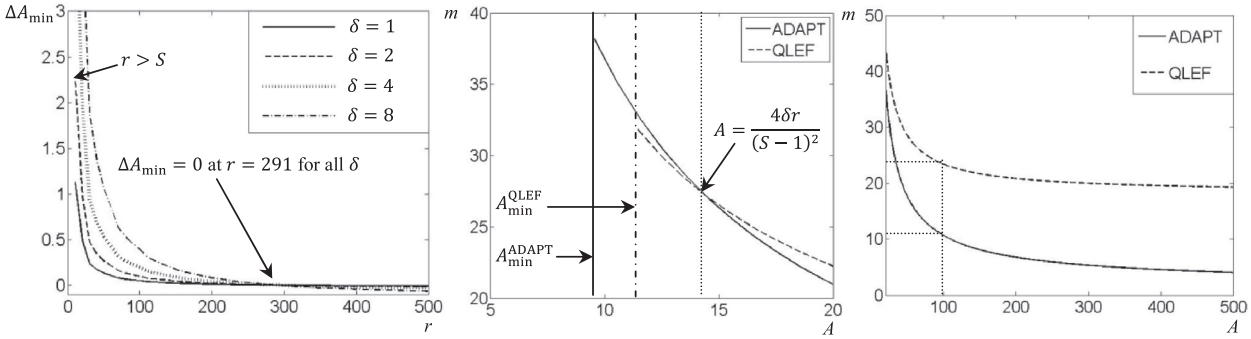
For the overall cost minimization scenario with $C = m + \tau A$ in (11), it is difficult to summarize the gap ratio in a closed-form expression as in Theorems 3–4, mainly because we can only get numerical solutions for (14). Never-theless, the gap ratio in this scenario will be demonstrated in the next Section.

4. Numerical analysis

Fig. 2 shows the operation zones and comparison of m for ADAPT and QLEF under a given packet delay bound. In particular, Fig. 2a shows that $\Delta A_{\min} = A_{\min}^{\text{QLEF}} - A_{\min}^{\text{ADAPT}}$ can be at most a few timeslots and this happens only when r is relatively small (e.g., $r < 50$). $\Delta A_{\min}$ becomes negative and trivial for $r > 291$. As a result, the zone $A_{\min}^{\text{QLEF}} > A \geq A_{\min}^{\text{ADAPT}}$ only exists for $r \leq 291$, where ADAPT must be used according to Theorem 1 since QLEF is infeasible. This is also shown in Fig. 2b with $r = 32$, where the dashed QLEF curve does not ex-ist in $A^{\text{QLEF}}{}_{\min} > A \geq A^{\text{ADAPT}}{}_{\min}$. Nevertheless, the minimum delay achieved by ADAPT and QLEF has no big difference from each other for all r, and the zone between $A_{\min}^{\text{QLEF}}$ and $A_{\min}^{\text{ADAPT}}$ is with small size as shown in both Fig. 2a and b. Note that $r \geq 6$ in Fig. 2a because QLEF must meet $r > S$.

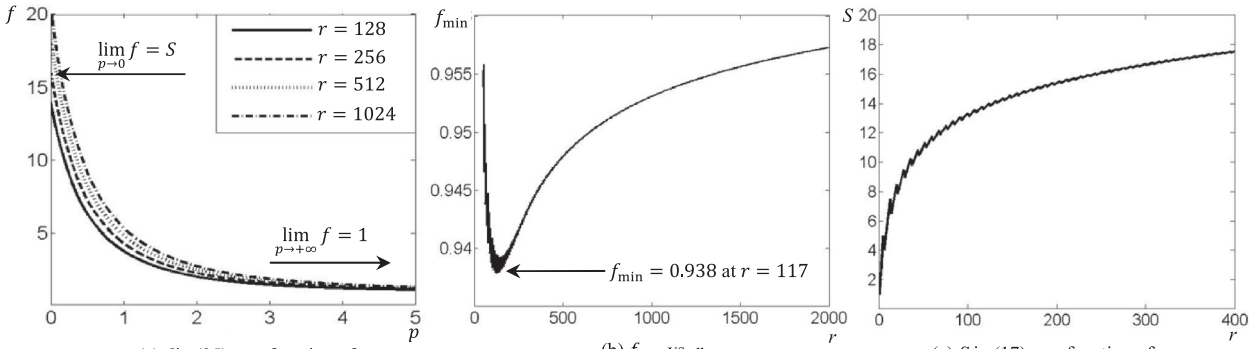
If $A \geq \max\{A_{\min}^{\text{ADAPT}}, A_{\min}^{\text{QLEF}}\}$, both ADAPT and QLEF are feasible, and Theorem 2 gives the branch point $4\delta r/(S-1)^2$ for choosing an algorithm to minimize m under a given A, as illustrated in Fig. 2b. It can be observed in Fig. 2b that ADAPT should be adopted for $A \geq 4\delta r/(S-1)^2$ and the gap on m increases with A. When A is larger than 50 or 100 timeslots, m in ADAPT will be much smaller than that in QLEF as shown in Fig. 2c. For example, if the number of racks in a DCN is $r = 512$ and the reconfiguration overhead is $\delta = 1$, ADAPT only needs about $m = 10$ middle switches for $A = 100$ and QLEF needs about $m = 23$. In either case, m is much smaller than r. As A increases, m decreases much faster when A is small, but keeps quite steady when A is large. Fig. 2c tells us that, by allowing a tolerable packet delay A, the number of middle switches m can be significantly reduced, but further enlarging A may not be so effective in cutting down m.

Fig. 3 illustrates the gap on m between the two algorithms based on f defined in (35). In particular, Fig. 3a provides an illustrative support to Theorem 3. Based on Theorem 4 (which focuses on a given r), we change r as in Fig. 3b to check the best performance that QLEF can possibly achieve as compared with ADAPT (denoted by $f_{\min}$ in (39)) under various values of A. It is revealed in Fig. 3b that m required by ADAPT can be at most 6.6% above that by QLEF, and this happens at $r = 117$ with $p = S$ as pointed out in Theorem 4. Note that there are some performance fluctuations around $r = 117$ in Fig. 3b. This is due to the fluctuations of S in the small r re-gion as shown in Fig. 3c, which is resulted from the roof and floor operations as well as the max function in (2).



(a) $\Delta A_{\min} = A_{\min}^{\text{QLEF}} - A_{\min}^{\text{ADAPT}}$ vs. r and δ. (b) Feasible and operation zones ($r = 32$, $\delta = 8$). (c) Gap on m ($r = 512$, $\delta = 1$).

Fig. 2. Operation zones of ADAPT and QLEF based on a given delay (A) for switch cost (m) minimization.



(a) f in (35) as a function of p. (b) $f_{\min}$ vs. r. (c) S in (17) as a function of r.

Fig. 3. Gap on m between ADAPT and QLEF for switch cost (m) minimization.

We now consider overall cost minimization by taking the cost of packet delay into account with C in (11). In this case, the value of A can be different in comparing ADAPT and QLEF as long as the overall cost can be minimized in each algorithm. Similar to (35), we define a ratio F in (40) to denote the relative performance of the two algorithms, where C_{ADAPT} and C_{QLEF} are the overall cost in ADAPT and QLEF, respectively.

$$F = \frac{C_{\text{ADAPT}}}{C_{\text{QLEF}}}. \tag{40}$$

Fig. 4 shows how F changes with τ and r based on our numerical experiments, where $F < 1$ means that ADAPT outperforms QLEF. For $r = 128$, it is observed that QLEF may slightly outperform ADAPT in a small-size zone of τ , and QLEF performance decays soon for large τ . As r becomes larger, QLEF can slightly outperform ADAPT in a larger zone of τ , and its performance decays for large τ in a smoother manner. A very interesting observation in Fig. 4 is that, each curve reaches its peak (i.e., maximum F) at

$$\tau = \frac{(r-S)^2}{\delta r}, \tag{41}$$

which is exactly the boundary value as formulated in (29). This is observed from our numerical experiments rather than theoretical analysis, because at present we can only use numerical methods to solve (14) for p instead of having a closed-form expression for more in-depth theoretical proof.

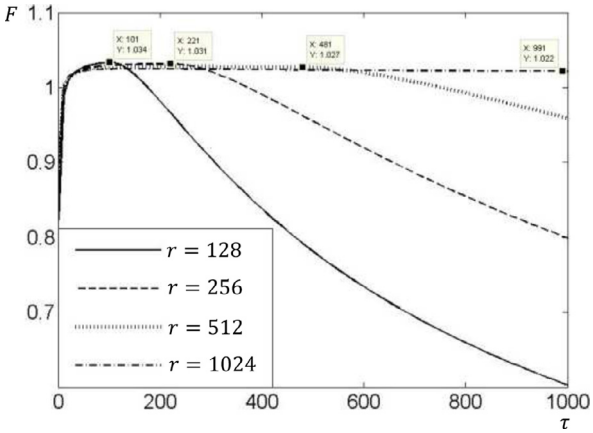


Fig. 4. ADAPT & QLEF comparison for overall cost minimization ($\delta = 1$).

5. Implementation issues and discussions

5.1. Traffic matrix decomposition techniques

So far, we have derived theoretical analysis based on ADAPT [21] and QLEF [20] algorithms without giving de-tails of the specific traffic matrix decomposition processes. Though such details can be readily referred to [20–21] and may not be necessary for carrying out theoretical analysis, they are indeed important in engineering practice. Besides,

ADAPT Matrix Decomposition Under a Given N_S

Input:
An $r \times r$ matrix $\boldsymbol{B} = \{b_{ij}\}$ with maximum line sum no more than nA, and the number of permutations N_S.
Output:
At most N_S permutations $\boldsymbol{P_k}$ and the corresponding weights φ_k.

Step 1. Calculate the quotient matrix $\boldsymbol{Q}$:
Construct an $r \times r$ matrix $\boldsymbol{Q} = \{q_{ij}\}$ such that

$$\boldsymbol{B} = \left\lfloor \frac{nA}{N_S - r} \right\rfloor \times \boldsymbol{Q} + \boldsymbol{R} \quad \text{and} \quad q_{ij} = \left\lfloor \frac{b_{ij}}{\lceil nA/(N_S - r) \rceil} \right\rfloor.$$

Step 2. Color $\boldsymbol{Q}$:
Construct a bipartite multigraph $\boldsymbol{G_Q}$ from $\boldsymbol{Q}$. Rows and columns of $\boldsymbol{Q}$ translate to two sets of vertices $\boldsymbol{X}$ and $\boldsymbol{Y}$ in $\boldsymbol{G_Q}$, and each entry $q_{ij} \in \boldsymbol{Q}$ translates to q_{ij} edges connecting vertices $i \in \boldsymbol{X}$ and $j \in \boldsymbol{Y}$. Find a minimal edge-coloring of $\boldsymbol{G_Q}$ to get at most $N_S - r$ colors, such that the edges incident on the same vertex have different colors. Set $1 \rightarrow k$.
Step 3. Schedule the quotient matrix $\boldsymbol{Q}$:
For a specific color in the edge-coloring of $\boldsymbol{G_Q}$, construct a permutation $\boldsymbol{P_k}$ from the edges in that color by setting the corresponding entries to 1 in $\boldsymbol{P_k}$ (all other entries in $\boldsymbol{P_k}$ are set to 0). Set the weight

$$\varphi_k = \left\lfloor \frac{nA}{N_S - r} \right\rfloor$$

and $k + 1 \rightarrow k$. Repeat Step 3 for each color in the edge-coloring of $\boldsymbol{G_Q}$.
Step 4. Schedule the residue matrix $\boldsymbol{R}$:
Find an arbitrary set of r non-overlapping permutations $\boldsymbol{P_k}$, $k \in \{N_S - r + 1, \cdots, N_S\}$ and set the following weight for each of them.

$$\varphi_k = \left\lceil \frac{nA}{N_S - r} \right\rceil.$$

Fig. 5. ADAPT matrix decomposition under a given N_S.

QLEF Matrix Decomposition

Input:
An $r \times r$ matrix $\boldsymbol{B} = \{b_{ij}\}$ with maximum line sum no more than nA.
Output:
$r \times r$ permutations $\boldsymbol{P}_1, \cdots, \boldsymbol{P}_r$ and weights $\varphi_1, \ldots, \varphi_r$.

Step 1: Initialization:
Set $0 \rightarrow k$. Initialize $\boldsymbol{P}_1, \cdots, \boldsymbol{P}_r$ to all-zero matrices and an $r \times r$ reference matrix $\boldsymbol{R} = \{r_{ij}\}$ to all 1*s*.
Step 2: Construct the first "half" configurations:
1) Un-shadow all lines (rows and columns) in $\boldsymbol{B}$ and $\boldsymbol{R}$. Set $1 \rightarrow w$.
2) Select the largest entry b_{ij} (with r_{ij}=1) in the not-yet-shadowed part of $\boldsymbol{B}$. If w=1, set $0 \rightarrow w$ and $b_{ij} \rightarrow \varphi_{k+1}$ where φ_{k+1} is the weight of $\boldsymbol{P}_{k+1}$. Shadow row i and column j in both $\boldsymbol{B}$ and $\boldsymbol{R}$, and set b_{ij} and r_{ij} to 0. Set $1 \rightarrow p_{ij}^{k+1}$ where p_{ij}^{k+1} is the entry (i, j) of $\boldsymbol{P}_{k+1}$. Repeat this step until $r - (2k + 1)$ entries are selected.
3) Construct a bipartite graph $\boldsymbol{U_G}$ from the remaining not-yet-shadowed part of $\boldsymbol{R}$ and perform maximum-size matching in $\boldsymbol{U_G}$ to get $2k + 1$ edges. Record their corresponding entries (i, j) to $\boldsymbol{P}_{k+1}$ by setting $1 \rightarrow p_{ij}^{k+1}$. Set these entries to 0 in both $\boldsymbol{B}$ and $\boldsymbol{R}$. Then set $k + 1 \rightarrow k$.
4) Repeat Steps 2.1-2.3 until $k = \lceil r/2 \rceil - 1$.
Step 3: Construct the second "half" configurations:
1) Un-shadow $\boldsymbol{B}$ and $\boldsymbol{R}$. Find the largest entry b_{ij} in the updated $\boldsymbol{B}$ and set b_{ij} as the weight for all the subsequent permutations.
2) Find a maximum-size matching in the bipartite graph of $\boldsymbol{R}$ and set the corresponding entries of $\boldsymbol{P}_{k+1}$ to 1. Set these entries to 0 in $\boldsymbol{B}$ and $\boldsymbol{R}$, and then set k+1$\rightarrow k$. Repeat this step until $k = r$.

Fig. 6. QLEF matrix decomposition.

the original ADAPT algorithm in [21] needs to be tailored, since we only need the matrix decomposition technique but determine the best N_S in this work using a different mechanism. As such, we briefly summarize ADAPT and QLEF matrix decomposition techniques in Figs. 5 and 6 respectively to make the paper fully self-contained. Note that φ_k in (2) is for the *worst-case* to ensure performance guaranteed switching for an arbitrary traffic matrix $\boldsymbol{B}$, and φ_k in Fig. 6 can be smaller for a specific $\boldsymbol{B}$.

5.2. Solution for the packet out-of-order problem

Our proposed scheme allows multi-path switching of the packets across multiple parallel middle switches. Generally, multi-path routing and switching may lead to the packet out-of-order problem in a flow. Though this may not be a big issue if a sequence number is piggybacked in the packet header, the switch network should not disorder the packets anyway.

Due to the crossbar nature of the middle switches, packet out-of-order problem does not exist for packets switched across the same middle switch. If the packets in a flow are simultaneously switched by different middle switches, they will go to different input ports of the same output switch (see Fig. 1). This provides an additional space domain to distinguish the order of packets coming from different middle switches. As a result, the packet out-of-order problem can be easily solved.

5.3. Multiplexing factor and speedup

In our system model, a high-speed optical interconnect with a multiplexing factor $M = n$ is assumed to connect each pair of ports between ToR and middle switches, which can transmit packets at a rate M times faster than the data rate of a server due to optical multiplexing. This assumption exactly matches the current situation in DCNs, where the low-rate server-level traffic is multiplexed onto a fiber for high-speed optical transmission. Though M is assumed as n in this work for simplicity, it can indeed take other values, which translates to introducing a constant factor into our theoretical analysis.

Theoretically, a multiplexing factor M at the ToR can be taken as a speedup of M. It ensures that the nA packets accumulated over A timeslots at each ToR switch can be transmitted across the middle switches in A timeslots. Other than the optical multiplexing, the middle switches do not need any frequency domain speedup in our scheme.

One may argue that, with a multiplexing factor or a speedup of M at the ToR switches, the conventional Clos network can use a less number of middle switches as well. Basically, we agree with this statement. However, existing works on Clos do not consider the reconfiguration overhead of middle switches, and the efficiency of traffic scheduling is another issue. It is interesting to ask how many middle switches could possibly be saved if a speedup is allowed at the ToRs. Based on in-depth theoretical analysis, our work gives a quantitative answer to this problem by taking the reconfiguration overhead into account.

5.4. Factors affecting the accuracy of the theoretical analysis

The accuracy of our theoretical analysis may be trivially affected by some factors. The first one is the *truncation error* as mentioned in Section 3.2, which is shown by a naive example in Fig. 7 with seven configurations averaged over three middle switches. In Scheme I (see Fig. 7), switching time of

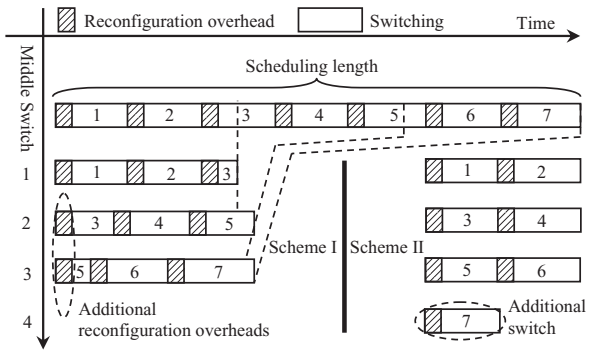


Fig. 7. Truncation error in engineering practice.

configurations 3 and 5 are divided and fulfilled by two middle switches. This leads to two more reconfiguration overheads with a biased averaging result. In contrast, Scheme II fulfills each configuration by the same middle switch, but requires one more middle switch. This naive example is just to show how truncation error can occur and can be treated in engineering. In practice, the scheduling length is generally large and is contributed by many configurations. As a result, the bias caused by the truncation error is generally trivial.

Truncation error also depends on the matrix decomposition techniques. In ADAPT, each permutation is equally weighted (see (1)), leading to the same switching time for all configurations. If SRF [21] or QLEF is adopted for matrix decomposition, unequal weights will be generated (see (2) for QLEF) and the situation will be more complicated. Though we can average the scheduling length in theoretical analysis as in (3) and (15), in practice we need to find the minimum feasible number of middle switches m to fulfill the unequally weighted permutations, where m is around our theoretical result but is slightly biased due to the truncation error. To this end, the problem matches the classical *bin-packing prob-lem* [22], where the middle switches translate to bins with uniform capacity and the unequally weighed permutations translate to items with different sizes no more than the bin capacity. The problem is to find the minimum number of bins to carry all items.

Other than truncation error, the accuracy of the theoreti-cal analysis may also be trivially affected by the integer roof and floor operations on N_S, m, A and φ_k, due to the intrinsic integer nature of those parameters.

6. Conclusion

We designed the switch network in DCNs (data center networks) with OpenFlow ToR switches and crossbar mid-dle switches using batch scheduling based packet switching, where each middle switch has a reconfiguration overhead. It was revealed that a tradeoff exists between switch cost and packet delay. By decomposing the traffic matrix into a set of permutations and fulfilling the permutations in parallel using the middle switches, multi-path switching is enabled and performance guaranteed switching is ensured among the DCN servers with a bounded packet delay and no packet loss. Based on the tradeoff, we minimized the switch cost as denoted by the number of middle switches under a given packet delay bound, and an overall cost metric by translating delay into a comparable cost factor. Criteria for choosing a proper matrix decomposition technique were also derived with insights demonstrated by numerical analysis. In the proposed scheme, the number of middle switches can be determined in a flexible manner to minimize the cost of the switch network, and can be significantly reduced by allowing a tolerable packet delay increase. Future work may study how the proposed scheme (which is based on batch scheduling and thus needs a packet accumulation time) can affect the overall end-to-end delay of DCN internal flows, as well as considering a different DCN internal connection topology and switching architecture. Distributed traffic scheduling is another interesting research topic as well.

Acknowledgments

This work is supported by the Major State Basic Research Program of China (973 project No. 2013CB329301 and 2010CB327806), the Natural Science Fund of China (NSFC project No. 61372085, 61032003, 61271165 and 61202379), and the Research Fund for the Doctoral Program of Higher Education of China (RFDP project No. 20120185110025, 20120185110030 and 20120032120041). It is also supported by Tianjin Key Laboratory of Cognitive Computing and Appli-cation, School of Computer Science and Technology, Tianjin University, Tianjin, P.R. China.

References

[1] K. Chen, C. Guo, H. Wu, J. Yuan, Z. Feng, Y. Chen, S. Lu, W. Wu, DAC: generic and automatic address configuration for data center networks, IEEE/ACM Trans. Network. 20 (1) (2012) 84–99.
[2] M. Bari, R. Boutaba, R. Esteves, L. Granville, M. Podlesny, M. Rabbani, Q. Zhang, M. Zhani, Data center network virtualization: a survey, IEEE Commun. Surv. Tutorials (2012) 1–20 pre-published.
[3] C. Lam, H. Liu, B. Koley, X. Zhao, V. Kamalov, V. Gill, Fiber optic communication technologies: what's needed for datacenter network operations, IEEE Commun. Mag. 48 (7) (2010) 32–39.
[4] H.J. Chao, Z. Jing, S.Y. Liew, Matching algorithms for three-stage bufferless Clos network switches, IEEE Commun. Mag. 41 (10) (2003) 46–54.
[5] S. Jiang, G. Hu, S.Y. Liew, H.J. Chao, Scheduling algorithms for shared fiber-delay-line optical packet switches-part II: the three-stage clos-network case, IEEE/OSA J. Lightw. Technol. 23 (4) (2005) 1601–1609.
[6] F. Wang, M. Hamdi, Strictly non-blocking conditions for the central-stage buffered Clos-network, IEEE Commun. Lett. 12 (3) (2008) 206–208.
[7] A. Jajszczyk, Nonblocking, repackable, and rearrangeable Clos networks: fifty years of the theory evolution, IEEE Commun. Mag. 41 (10) (2003) 28–33.
[8] F.K. Hwang, W.-D. Lin, V. Lioubimov, On noninterruptive rearrangeable networks, IEEE/ACM Trans. Network. 14 (5) (2006) 1141–1149.
[9] X. Yuan, W. Nienaber, Z. Duan, R. Melhem, Oblivious routing in fat-tree based system area networks with uncertain traffic demands, IEEE/ACM Trans. Network. 17 (5) (2009) 1439–1452.
[10] S. Coll, F.J. Mora, J. Duato, F. Petrini, Efficient and scalable hardware-based multicast in fat-tree networks, IEEE Trans. Parallel Distrib. Syst. 20 (9) (2009) 1285–1298.
[11] F.O. Sem-Jacobsen, T. Skeie, O. Lysne, J. Duato, Dynamic fault tolerance in fat trees, IEEE Trans. Comput. 60 (4) (2011) 508–525.
[12] L. Liu, D. Zhang, T. Tsuritani, R. Vilalta, R. Casellas, L. Hong, I. Morita, H. Guo, J. Wu, R. Martinez, R. Munoz, Field trial of an OpenFlow-based unified control plane for multi-layer multi-granularity optical switching networks, IEEE/OSA J. Lightw. Technol. (2012) pre-published.
[13] H. Jin, D. Pan, J. Liu, N. Pissinou, OpenFlow based flow-level band-width provisioning for CICQ switches, IEEE Trans. Comput. (2012) pre-published.
[14] G. Antichi, A. Di Pietro, S. Giordano, G. Procissi, D. Ficara, Design and development of an OpenFlow compliant smart gigabit switch, IEEE GLOBECOM 2011 (2011).

[15] G. Birkhoff, Tres observaciones sobre el algebra lineal, Univ. Nac. Tucumán Rev. Ser. A 5 (1946) 147–151.
[16] J. von Neumann, A certain zero-sum two-person game equivalent to the optimal assignment problem, Contributions to the Theory of Games, 2, Princeton Univ. Press, Princeton, New Jersey, 1953, pp. 5–12.
[17] J. Li, N. Ansari, Enhanced Birkhoff-von Neumann decomposition algorithm for input queued switches, IEE Proc-Commun 148 (6) (2001) 339–342.
[18] B. Towles, W.J. Dally, Guaranteed scheduling for switches with configuration overhead, IEEE/ACM Trans. Network. 11 (5) (2003) 835–847.
[19] X. Li, M. Hamdi, On scheduling optical packet switches with reconfiguration delay, IEEE J. Selected Areas Commun. 21 (7) (2003) 1156–1164.
[20] B. Wu, K.L. Yeung, P.-H. Ho, X.H. Jiang, Minimum delay scheduling for performance guaranteed switches with optical fabrics, IEEE/OSA J. Lightw. Technol. 27 (16) (2009) 3453–3465.
[21] B. Wu, K.L. Yeung, M. Hamdi, X. Li, Minimizing internal speedup for performance guaranteed switches with optical fabrics, IEEE/ACM Trans. Network. 17 (2) (2009) 632–645.
[22] M. Maiza, A. Labed, M.S. Radjef, Efficient algorithms for the offline variable sized bin-packing problem, J. Global Opt. (2012).