Cost and Delay Tradeoff in Three-Stage Switch Architecture for Data Center Networks

Shu Fu†, Bin Wu†, Xiaohong Jiang†, Achille Pattavina†, Lei Zhang‡ and Shizhong Xu†

† School of Computer Science and Technology, Tianjin University, Tianjin, P. R. China (emails: fshfs@hotmail.com.cn, binwu.tju@gmail.com, lizhang@tju.edu.cn)
‡ School of Systems Information Science, Future University Hakodate, Hakodate, Japan (email: jiang@fun.ac.jp)
§ Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy (email: pattavina@elet.polimi.it)

Abstract—Data center networks (DCNs) generally adopt Clos network with crossbar middle switches to achieve non-blocking data switching among the servers, and the number of middle switches is proportional to the number of ports of the aggregation switches in a fixed manner. Besides, reconfiguration overhead of the switches is generally ignored, which may contradict the engineering practice. In this paper, we consider batch scheduling based packet switching in DCNs with reconfiguration overhead at each middle switch, which inevitably leads to packet delay. With existing state-of-the-art traffic matrix decomposition algorithms, we can generate a set of permutations, each of which stands for the configuration of a middle switch. By reconfiguring each middle switch to fulfill multiple configurations in parallel with others, we reveal that a tradeoff exists between packet delay and switch cost (denoted by the number of middle switches), while performance guaranteed switching with bounded packet delay can be achieved without any packet loss. Based on the tradeoff, we can minimize the number of middle switches (under a given packet delay bound) and an overall cost metric (by translating delay into a comparable cost factor), as well as formulating criterions for choosing a matrix decomposition algorithm. This provides a flexible way to reduce the number of middle switches by slightly enlarging the packet delay bound.

Keywords-Clos network, Data center networks (DCNs), matrix decomposition, packet switching, scheduling.

I. INTRODUCTION

DCNs (Data Center Networks) has attracted great attentions as the enabling technology for cloud services. Servers in a DCN are organized into racks, and exchange data via a multi-stage switch network such as Clos [1-2] or Fat-Tree [3]. Fig. 1 shows a three-stage Clos network Clos(n, m, r). It has r input switches (n × m), m middle switches (r × r) and r output switches (m × n), where the kth switch in a stage has a link connected to the kth input port of each and all switches in the next. In terms of DCNs, the rth pair of input and output switches in Fig. 1 can be combined into a ToR (Top-of-Rack) switch at the kth rack, which can be routers or OpenFlow switches [4] capable of local in-rack data switching. On the other hand, cross-rack traffic is handled by the middle switches which are generally crossbars.

Existing works on Clos network [2, 5] focus on achieving non-blocking communications. It is shown that the required number of middle switches m is proportional to the number of ports of the ToR switches in a fixed manner (e.g. m ≥ n for rearrangeably non-blocking [5]). In Clos based DCNs, m is critical in determining the cost of the switch network. Since m equals to the number of output ports of each ToR switch, a larger m leads to more output ports of each ToR switch, and complicates the DCN interconnects. Besides, middle switches could be expensive as well if they are high-capacity optical switches. To cut down the system cost and make the DCN more scalable, it is desirable to reduce the number of middle switches.

In this paper, we focus on three stage switch architecture as in Fig. 1 for DCNs, with OpenFlow ToR switches [4] and high-capacity optical crossbar middle switches. At the ToR, low-rate traffic of the in-rack servers can be multiplexed onto fibers for high-speed optical transmission to the middle (or core) switches. Though an additional aggregation stage may exist between ToR and core switches in practical DCNs, it is treated as the input or output stage in Fig. 1. Such an architecture is with much reduced cost and power consumption. But, each optical switch requires a reconfiguration overhead to adjust its tunable optical components. In multi-stage switching, the ToR and core switches also need additional time to be synchronized. During the reconfiguration overhead time, switching is idle and no data can be transmitted across the middle switches. As a result, packet delay is inevitable.

Under the above architecture, we consider batch scheduling based packet switching [6-10] with reconfiguration overhead at the middle switches. By accumulating packets over a period of time at the input switches, a traffic matrix B is obtained as a batch to denote the aggregated traffic between an arbitrary pair of input and output switches. Our approach is to decompose B into a set of permutations, each of which denotes a crossbar configuration. By
reconfiguring each middle switch to fulfill multiple configurations, parallel multi-path switching is achieved at the middle switches. With the existing traffic matrix decomposition algorithms [9-10], our work ensures performance guaranteed switching without packet loss and with a bounded packet delay.

Our focus is on the tradeoff between the packet delay and \( m \) as in Fig. 1, and minimizing the cost of the switch network, as well as formulating criteria for choosing a matrix decomposition algorithm to achieve cost minimization. Intuitively, a larger \( m \) leads to a smaller packet delay due to more switching paths to shorten the batch switching time. That is why a tradeoff exists and \( m \) can possibly be reduced at the cost of a slightly larger packet delay bound. In our work, the constraint \( m \geq n \) is removed. This makes the switch network design in DCNs more flexible.

II. SYSTEM MODEL AND PROBLEMS

A. System Model

Our system model is based on the three-stage Clos\((n,m,r)\) with OpenFlow [4] ToR switches (combined input and output switches in Fig. 1), which are capable of collecting/distributing packets from/to the servers in the rack by proper port matching. The ToR switches are buffered, such that outgoing packets of the racks can be accumulated over time to form a batch, 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.

The middle switches are optical crossbars for handling cross-rack traffic, each with a reconfiguration overhead of \( \delta \) timeslots. The connection status of a middle switch is called a configuration and is denoted by an \( r \times r \) permutation. 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. 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 \( B \) 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. The total number of packets destined to each output port is assumed to be no more than \( A \), and that arrived at each output switch is at most \( nA \). Each row of \( B \) matches an input switch and each column an output switch. An entry in \( B \) denotes the number of packets to be transmitted between the corresponding switch pair. Accordingly, the maximum line (row or column) sum of \( B \) equals to \( nA \). The \( nA \) packets at a ToR switch can be transmitted by a high-speed optical interconnect in \( A \) timeslots since \( M = n \) is assumed in this work. This makes our model exactly the same as those in [7-10], 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). In what follows, we slightly abuse \( A \) as the packet delay (bound) for simplicity.

B. 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 matrix decomposition technique (either ADAPT [9] or QLEF [10]) 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 \( r \) as a per-unit-delay cost to translate packet delay into a cost factor comparable with the switch cost, which allows us to combine both factors 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 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.

III. TRADEOFF AND COST MINIMIZATION

The tradeoff between switch cost (denoted by \( m \)) and packet delay can be formulated under a specific matrix decomposition technique. Our work is based on ADAPT [9] and QLEF [10] decompositions since they are typical representatives among those in [7-10]. In what follows, we first briefly summarize the theoretical aspects of the two algorithms, and then formulate the tradeoff and cost minimization as well as the criteria for choosing a proper matrix decomposition technique. Note that the detailed processes of ADAPT and QLEF decompositions are indeed not important for theoretical analysis, and can be readily referred to [9-10] as plug-in modules for this work.

A. Summary of ADAPT and QLEF Algorithms

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

Given an \( r \times r \) traffic matrix \( B \) with a maximum line sum of \( nA \), ADAPT algorithm [9] can generate a set of \( N_5 \) \((r^2 - 2r + 2 > N_5 > r)\) permutations \( \{ P_k \} \) \((N_5 \geq k \geq 1)\) to cover \( B \), with each \( P_k \) equally weighted by

\[
\varphi_k = \frac{nA}{N_5 - r}. \tag{1}
\]

Note that \( N_5 = r \) matches QLEF [10] and \( N_5 = r^2 - 2r + 2 \) matches Birkhoff-von Neumann decomposition [6]. More specifically, ADAPT in [9] also determines the best \( N_5 \) value for

\[
\varphi_{k+1} \left| _{r \geq k+2} \right| = \max \left\{ \frac{nA}{k - \Delta + 1} \right\}, \quad \varphi_{k+1} \left| _{r \geq k+1} \right| = \frac{nA}{k - \Delta + 1}. \tag{2}
\]
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 [10] generates exactly \( N_S = r \) permutations \( \{ P_k \} \) to cover \( B \) with a worst-case weight \( \varphi_k \) in (2) for each \( P_k \).

**B. ADAPT Based Switch Cost and Packet Delay Tradeoff**

Assume that ADAPT [9] is used to decompose the traffic matrix \( B \) into \( N_S \) permutations. Due to the reconfiguration overhead \( \delta \) of the middle switches and the assumption of \( M = n \) (i.e., the multiplexing factor), we have

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

where \( m \) is the number of middle switches, and \( \varphi_k \) is the number of timeslots that a configuration \( P_k \) 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 overthrottled. 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 overhead 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 [7-10], which is the total number of timeslots for transmitting all packets in \( B \) 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 equally average the scheduling length over the \( m \) middle switches, while ensuring each configuration to be fulfilled only by a single switch. However, this is mainly an engineering concern and it can only trivially bias our theoretical results. Other than the truncation error, we do not allow a configuration to be separated and fulfilled by two middle switches.

Equations (1) and (3) lead to

\[
A = \frac{\delta N_s}{m - \frac{N_S}{A} - \frac{r}{\sqrt[3]{A^2}}}, \tag{4}
\]

\[
m = \frac{\delta N_s + N_S - r}{A + \frac{N_S}{A} - \frac{r}{\sqrt[3]{A^2}}}. \tag{5}
\]

It is clear in (4)-(5) that a larger \( A \) matches a smaller number \( m \) of middle switches, and vice versa.

**C. 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.

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

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

The inequality in (6) holds because \( N_S > r \) must be true in order to achieve performance guaranteed switching [7-10]. 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

We get

\[
\frac{d m}{d N_S} = \delta - \frac{r}{(N_S - r)^2} = 0. \tag{7}
\]

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

Note that integer \( N_S \) is treated as a real value in theoretical analysis. In practice, we can use \( [N_S] \) to replace \( N_S \) obtained in (8).

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

\[
m = \frac{\delta N_S}{A} + \frac{N_S}{A} - \frac{r}{\sqrt[3]{A^2}} = (1 + p)^2, \tag{9}
\]

where

\[
p = \sqrt{\frac{rA}{\delta}}. \tag{10}
\]

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).

**D. ADAPT Based Overall Cost Minimization with Delay Cost**

In Section III.C, 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 II.B, we can combine switch cost and packet delay into an integrated overall cost \( C \) as in (11), by defining a per-unit-delay cost \( \tau \) 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

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

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 III.C, we have

\[
\frac{d^2 C}{dA^2} = \frac{1}{A^3} \left( \frac{3\sqrt{\tau A}}{2} + 28r \right) > 0. \tag{13}
\]

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

\[
\frac{dC}{dA} = -\frac{\delta r}{A^2} - \frac{\sqrt{\tau}}{A^3 + \tau} = -\frac{1}{\sqrt{\tau r}} (p^4 + p^3 - \tau p) = 0. \tag{14}
\]

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 \( C \).

**E. QLEF Decomposition Based Results**

QLEF (Quasi Largest Entry First) algorithm is originally proposed in [10] to schedule traffic in a single switch with reconfiguration overhead, where it minimizes the packet delay bound using \( N_S = r \) configurations. With multiple middle switches in our case, the situation is quite different.
Since $N_S = r$ in QLEF, we assume $m \leq r$, and (3) leads to
\[
\frac{1}{m} \left( \frac{\delta r}{m - S} + \frac{1}{m} \sum_{k=1}^{r} \varphi_k \right) = A,
\] (15)
where $\varphi_k$ ($r \geq k \geq 1$) can be calculated in (2) [10] based on the system parameters $A$ and $r$. Equation (15) is equivalent to
\[
\frac{1}{m} \left( \frac{\delta r}{A} + \frac{1}{nA} \sum_{k=1}^{r} \varphi_k \right) = \frac{1}{m} \left( \frac{\delta r}{A} + S \right) = 1,
\] (16)
where
\[
S = \frac{1}{nA} \sum_{k=1}^{r} \varphi_k
\] (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 $m \leq r$.
\[
A = \frac{\delta r}{m - S};
\] (18)
\[
m = \frac{\delta r}{A} + S = p^2 + S.
\] (19)

Accordingly, the minimum achievable values of $A$ and $m$ are
\[
A_{\min}^{\text{QLEF}} = \frac{\delta r}{r - S};
\] (20)
\[
m_{\min}^{\text{QLEF}} > S;
\] (21)

Meanwhile, the overall cost $C$ in (11) translates to
\[
C = m + \tau A = p^2 + S + \tau A = \frac{\delta r}{A} + S + \tau A,
\] (22)
and thus
\[
\frac{d^2C}{dA^2} = \frac{2\delta r}{A^3} > 0.
\] (23)

Therefore, the value of $A$ for minimizing $C$ is the solution of
\[
\frac{dC}{dA} = \tau - \frac{\delta r}{A^2} = 0,
\] (24)
which is
\[
A = \sqrt{\frac{\delta r}{\tau}}.
\] (25)

From (19), (22) and (25), the minimum overall cost $C$ and the achieving number of the middle switches $m$ are
\[
m = \frac{\delta r}{A} + S|_{A = \sqrt{\frac{\delta r}{\tau}}} = \sqrt{\delta r \tau} + S;
\] (26)
\[
C = m + \tau A|_{A = \sqrt{\frac{\delta r}{\tau}}} = 2\sqrt{\delta r \tau} + S.
\] (27)

Also note that $\tau$ must be properly constrained in order to use (25)-(27) to calculate $A$, $m$ and $C$ in the QLEF scenario. This is because $m \leq r$ entails
\[
m = \sqrt{\delta r \tau} + S \leq r.
\] (28)

Therefore, the following (29) must be satisfied.
\[
\tau \leq \frac{(r - S)^2}{\delta r}.
\] (29)

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

**F. Operation Zones of ADAPT and QLEF**

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

**Proof:** If $A < A_{\min}^{\text{QLEF}}$, QLEF is infeasible by our assumption. Instead, ADAPT can be feasible according to (8)-(10), though $m$ could be relatively large. On the other side, the assumption of $r \geq m$ and (21) in QLEF lead to $r > S$. The remaining issue is to prove that the minimum achievable $A$ in ADAPT is
\[
A_{\min}^{\text{ADAPT}} = \frac{4\delta r}{\sqrt{1 + 4\delta r} - 1}.
\] (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, 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
\[
(1 + p)^2 = r + \frac{r}{p}
\] (31)
and
\[
p = \frac{\sqrt{\frac{\delta r}{A}}}{\sqrt{\frac{\delta r}{A} + \frac{\delta r}{5} - 1}}.
\] (32)

Note that $A = A_{\min}^{\text{ADAPT}}$ 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 $4\delta r/(S - 1)^2 > A \geq \max\{A_{\min}^{\text{QLEF}}, A_{\min}^{\text{ADAPT}}\}$ with $S$ in (17) and $\varphi_k$ in (2). Otherwise, ADAPT should be adopted if $A \geq 4\delta r/(S - 1)^2$.

**Proof:** When $A \geq \max\{A_{\min}^{\text{QLEF}}, A_{\min}^{\text{ADAPT}}\}$, both ADAPT and QLEF are feasible, but $4\delta r/(S - 1)^2 > A$ entails
\[
(1 + p)^2 = r + \frac{r}{p}
\] (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.

\[
f = \frac{p^2 + S}{(1 + p)^2}.
\] (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 $A$, equation (36) holds in the zones where both ADAPT and QLEF are feasible.
\[
\frac{p}{p \to +\infty} \frac{\tau}{\tau \to 0} \quad \text{and}\quad \frac{p}{p \to +\infty} \frac{\varphi_k}{m} - \frac{m}{m \to +\infty} \quad \text{and}\quad \frac{p}{p \to +\infty} \frac{\varphi_k}{m}.
\] (36)

Theorem 3 tells us that if $p \to +\infty$ (e.g., $r \to +\infty$ with given $\delta$ 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 $\delta$ 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

\[
\frac{df}{dp} = \frac{2(p - S)}{(1 + p)^3} = 0,
\]

we have \( p = S \). Moreover, we have

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

which is positive at \( p = S \). Consequently, \( f \) is minimized as

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

at \( p = S \).

**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 \( r \) satisfies (29), ADAPT decomposition should be used if \( C(A) < 2\sqrt{4rT + S} \) and QLEF should be used otherwise. When (29) is not satisfied, this condition changes to \( C(A) < r + \delta r / (r - S) \).

**Proof:** When \( r \) satisfies (29), the minimum overall cost \( C \) in QLEF can be calculated in (27) as \( C = 2\sqrt{4rT + S} \). Otherwise, \( A \) in QLEF will be \( A_{\min}^{QLEF} \) as in (20) with \( m = r \), which leads to \( C = m + \tau A = r + \delta r / (r - S) \). In either case, ADAPT should be used if \( C(A) < C \) and QLEF otherwise.

**IV. 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 \( \Delta A_{\min} = A_{\min}^{QLEF} - A_{\min}^{ADAPT} \) 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 \( A_{\min}^{QLEF} > A \geq A_{\min}^{ADAPT} \) 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 exist in \( A_{\min}^{QLEF} > A \geq A_{\min}^{ADAPT} \). Nevertheless, the minimum delay achieved by ADAPT and QLEF has no big difference from each other for all \( r \), and the zone between \( A_{\min}^{QLEF} \) and \( A_{\min}^{ADAPT} \) is with small size as shown in both Fig. 2a and Fig. 2b. Note that \( r \geq 6 \) in Fig. 2a because QLEF must meet \( r > S \).

If \( A \geq \max \{ A_{\min}^{ADAPT}, A_{\min}^{QLEF} \} \), 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 shows the gap on \( m \) between the two algorithms based on \( f \) in (35). In particular, Fig. 3a supports 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. There are some performance fluctuations around \( r = 117 \) in Fig. 3b. This is due to the fluctuations of \( S \) in the small \( r \) region as shown in Fig. 3c, which is resulted from the roof and floor operations as well as the max function in (2).

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 \( C_{\text{ADAPT}} \) and \( C_{\text{QLEF}} \) 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 \( \tau \) 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 \( \tau \), and QLEF performance decays soon for large \( \tau \). As \( r \) becomes larger, QLEF can slightly outperform ADAPT in a larger zone of \( \tau \), and its performance decays for large \( \tau \) 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.

**V. CONCLUSION**

We designed the switch network in DCNs (data center networks) with OpenFlow ToR switches and optical crossbar middle 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.

REFERENCES


