On Fast and Coordinated Data Backup in Geo-Distributed Optical Inter-Datacenter Networks

Jingjing Yao, Ping Lu, Long Gong, and Zuqing Zhu, *Senior Member, IEEE*

***Abstract*—In an optical inter-datacenter (inter-DC) network, for preventing data loss, a cloud system usually leverages multiple DCs for obtaining sufficient data redundancy. In order to improve the data-transfer efficiency of the regular DC backup, this paper investigates fast and coordinated data backup in geographically distributed (geo-distributed) optical inter-DC networks. By considering a mutual backup model, in which DCs can serve as the backup sites of each other, we study how to finish the regular DC backup within the shortest time duration (i.e., DC backup window (DC-B-Wnd)). Specifically, we try to minimize DC-B-Wnd with joint optimization of the backup site selection and the data-transfer paths. An integer linear programming (ILP) model is first formulated, and then we propose several heuristics to reduce the time complexity. Moreover, in order to explore the tradeoff between DC-B-Wnd and operational complexity, we propose heuristics based on adaptive reconfiguration (AR). Extensive simulations indicate that among all the proposed heuristics, AR-TwoStep-ILP achieves the best tradeoff between DC-B-Wnd and operational complexity and it is also the most time-efficient one.**

***Index Terms*—Backup window, datacenter backup, mutual backup model, optical inter-datacenter networks.**

I. Introduction

NOWADAYS, the rapid rise of Internet-scale cloud systems makes datacenter (DC) networking a hot topic [1], [2]. Cloud services delivered by DC networks provide huge opportunities for emerging applications such as online gaming, video on demand, social networking, collaborative computing, etc. It is known that customer experience is vital for these interactive applications, which normally only have very small tolerance to service disruptions. For instance, recent studies showed that customers start to abandon an online video if it takes more than 2 s to load, the abandonment rate will increase 5.8% for each incremental delay of 1 s, and 60% of the abandoners will never come back [3]. Therefore, large enterprises such as Google, Amazon and Microsoft, have been building DCs in geographically distributed (geo-distributed) locations to provide cloud services with quality-of-service and quality-of-experience guarantees [2]. Even though deploying DCs close to end-users reduces service latency, how to build and manage the network that interconnects these DCs is challenging as the traffic would exhibit coexistence of huge peak throughput and high burstiness [4]. Fortunately, optical networks have the advantages of huge capacity, low latency, high availability and low power consumption, and can work as viable substrate infrastructure for inter-DC networks [5], [6].

Meanwhile, inter-DC networks are vulnerable to natural disasters as earthquake, hurricane, tsunami, and tornado can wipe DCs out easily and cause long service interruption and huge data losses. For example, the 2008 Sichuan earthquake had devastated over 60 enterprise DCs [7], and the 2011 Tōhoku earthquake and tsunami had brought down tens of DCs and even made a few companies file bankruptcy [8]. Hence, in order to prevent data losses and obtain sufficient data redundancy, we need to leverage regular and periodic data backup among geo-distributed DCs [9], [10].

Note that DC backup may impact normal network services significantly, as it usually consumes huge bandwidth and can introduce prolonged network congestion. Therefore, it is desired to finish data backup within a relatively short time. Here, we define the time period for backing up new data in all the DCs of an inter-DC network (i.e., a regular DC backup) as DC backup window (DC-B-Wnd) [11], which is a key metric for evaluating the backup scenario. Apparently, a shorter DC-B-Wnd leads to shorter disruption on normal services, and thus makes the corresponding backup scenario more attractive. In order to shorten DC-B-Wnd, previous studies have tried to reduce the size of the data to be backed up with different data compression techniques, including de-duplication [12], snapshot [13], and data redundancy reduction [14].

We need to clarify that the regular DC backup discussed in this work is different from the data replication in [15], [16] DCs, which involves copying and moving data between virtual machines or servers to ensure data consistency and improve the reliability and accessibility [17]. Hence, data replication usually is application-specific and needs to be done in real or nearly-real time. On the other hand, in a regular DC backup, each DC needs to copy all the data that it has produced in a period to the backup site, for obtaining sufficient data redundancy. Since a DC can store various types of data and the amount of data to be backed up is usually huge, regular DC backup will not be application-specific or in real-time.

In addition to data compression, we can also optimize the DC backup scenario by selecting proper peer DC(s) as a DC's backup site(s) and calculating the corresponding data-transfer path(s). Therefore, inspired by the studies on advance

Manuscript received October 27, 2014; revised March 2, 2015 and April 19, 2015; accepted April 20, 2015. Date of publication April 21, 2015; date of current version June 3, 2015. This work was supported in part by the NCET Project NCET-11-0884, the NSFC Project 61371117, the Fundamental Research Funds for the Central Universities (WK2100060010), Natural Science Research Project for Universities in Anhui (KJ2014ZD38), and the Strategic Priority Research Program of the Chinese Academy of Sciences (XDA06030902).

The authors are with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China (e-mail: yao2jing@mail.ustc.edu.cn; lpbest@mail.ustc.edu.cn; gonglong@mail.ustc.edu.cn; zqzhu@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JLT.2015.2425303

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.2015.2425303, Journal of Lightwave Technology

reservation (AR) and sliding scheduling [18]–[21], this work leverages a discrete-time anycast network model and studies how to minimize DC-B-Wnd in optical inter-DC networks with joint optimization of backup site selection and data-transfer paths. We consider a mutual backup model [22] and assume that the network is a discrete-time system that operates on fixed time intervals (i.e., Δt) [23], [24]. Hence, in between two adjacent intervals, the DC backup scenario (i.e., backup site selection and data-transfer paths) can be re-optimized.

With this network model, we first formulate an integer linear programming (ILP) model with the objective to minimize DC-B-Wnd. As the ILP is intractable for large-scale problems, we then propose several heuristics to reduce the computational complexity. Our contributions are as follows.

1) We consider mutual backup in a discrete-time optical inter-DC network and study how to minimize DC-B-Wnd with joint optimization of backup site selection and data-transfer paths along the time axis.
2) We leverage dynamic anycast (i.e., the backup site(s) of a DC is flexible) and lightpath reconfiguration to maximize the throughput for DC backup.
3) We present an ILP model that can obtain the exact solution to minimize DC-B-Wnd. Specifically, based on the network status, the ILP model optimizes the scenario of a regular DC backup by determining the backup site selection and data-transfer paths for each time interval along the time axis.
4) To reduce DC-B-Wnd with consideration of the overhead from lightpath reconfigurations, we propose a heuristic, namely, FR-TwoStep-ILP. FR-TwoStep-ILP jointly considers all the DC backup pairs with two ILPs, both of which can be solved in polynomial time, and provides solutions that are very close to the optimal ones.
5) We design adaptive reconfiguration (AR) schemes (i.e., AR-based heuristics) to further reduce lightpath reconfigurations in DC backup, and in the meantime the performance on DC-B-Wnd will not be degraded too much.

The rest of the paper is organized as follows. Section II reviews the related works. We present the network model for DC backup in an optical inter-DC network in Section III. Section IV formulates the ILP model to minimize DC-B-Wnd. The time-efficient heuristics are proposed in Section V, and we discuss the performance evaluation with numerical simulations in Section VI. Finally, Section VII summarizes the paper.

II. Related Work

Generally speaking, there are two pillars for data protection in inter-DC networks, i.e., regular backup and failure recovery [25]. Regular backup usually happens when the network is in its normal state without failures, and each DC tries to distribute the newly-generated data on it to a remote backup site. Hence, regular backup enhances data redundancy and plans ahead for the network failures that can impact DCs. By doing so, it prevents the incident that data can never be restored due to DC failures. On the other hand, failure recovery leverages network protection/restoration schemes to maintain service continuity during network failures. For instance, the network operator can use node- and path-protection to make sure that customers can still access their data and applications when their working DC fails. Previously, for failure recovery, researchers have discussed the resilient network dimensioning for optical clouds in [26], and have considered the disaster-resilient network design in [27]. Since failure recovery with protection/restoration schemes has been well studied before, we will not address it in this work but focus on the network operation to achieve fast and coordinated regular backup.

The essential problem of regular DC backup is how to schedule the data transfers to minimize DC-B-Wnd. Andrei *et al.* proposed an interesting sliding scheduling scheme in [19], where they considered the lightpath requests whose start-time is not specified but can slide in a predefined time window. Note that the work in [19] only addressed the unicast requests that had specific sources and destinations and fixed bandwidth requirements. In [20], the authors studied the variable-bandwidth AR schemes, which still use the unicast model.

Even though sliding scheduling looks similar to the problem addressed in this work, there are still some fundamental differences between them. First of all, slide scheduling only considered the flow-oriented request model, in which a fixed or variable amount of bandwidth should be reserved for each request for a continuous period of time. Both studies in [19], [20] assumed that lightpath requests would be blocked if cannot be served within the maximum setup delay and they cannot be paused in their lifetime. However, we actually consider the data-oriented request for regular DC backup. Specifically, given an inter-DC network where each DC needs to back up a fixed amount of data to remote DC(s), we need to find a backup scenario based on the network status, which can accomplish all the backup-related data-transfers as soon as possible. The DC backup is done in a progressive way. Consequently, there is no request blocking in the network and we may pause a data-transfer when necessary. To this end, we can see the network models of slide scheduling and our problem are different. Moreover, since we use anycast and lightpath reconfiguration (i.e., as time goes on, the DC backup pairs and the data-transfer paths can change), our service provisioning scheme is more flexible, which makes the optimization more complex.

Previous investigations have also addressed bulk-data transfer in inter-DC networks [23], [24]. Nevertheless, they were for the one-to-one scheme in which the source and destination of each request were given, which is not the case for the DC backup in this work as we consider the mutual backup model and incorporate anycast and lightpath reconfiguration. In [22], we have studied how to minimize DC-B-Wnd with joint optimization of backup site selection and data-transfer paths, and showed some preliminary results. This work expands our previous work in [22]. Basically, we propose new algorithms and include more theoretical analysis and simulation results to make the work more comprehensive.

III. Network Model

We consider the scenario that the optical inter-DC network uses a wavelength-division multiplexing (WDM) network as the substrate infrastructure. The network operator has the

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.2015.2425303, Journal of Lightwave Technology

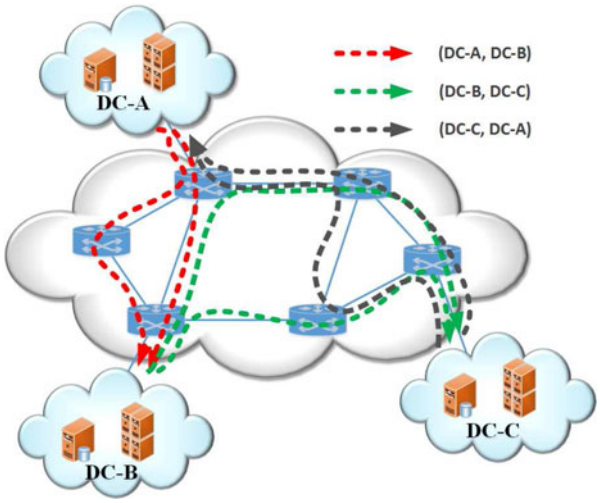

Fig. 1. Mutual backup in an optical inter-DC network.

network control and management (NC&M) capability to decide the DC backup scenario and establish the physical connections (i.e., lightpaths) to support the corresponding data-transfer. In practical networks, this can be done by converging the optical infrastructure with DC network [6] and leveraging transport-aware cross-stratum optimization with, e.g., software-defined networking [28]. For the lightpath provisioning, we allocate bandwidth on the optical fibers based on wavelength channels. In this work, we assume that the optical cross-connects (OXCs) in the WDM network has the capability of performing optical-to-electrical-to-optical (O/E/O) conversions, and hence we do not need to consider the wavelength continuity constraint when setting up the lightpaths.

Here, we address the regular DC backup, where each DC tries to distribute the new data that it generates within a period of time to remote DC(s) for obtaining sufficient data redundancy. Therefore, if something happens to bring it down, the missing data can be successfully restored from its backup site(s). The DC backup process uses the mutual backup model [22], which means that the DCs in the network can serve as the backup sites of one another. Specifically, we refer to a DC that has data to be backed up as the production DC, which needs to choose a remote DC as its backup site, and meanwhile, itself can also be chosen as the backup site of a remote DC. For instance, in Fig. 1, the inter-DC network includes three DCs, which form three backup pairs, i.e., {production DC, backup site}, as {*DC-A, DC-B*}, {*DC-B, DC-C*} and {*DC-C, DC-A*}. In order to ensure that the DC backup can be finished as soon as possible, we assume that multiple lightpaths can be set up to support the data-transfer for each backup pair, as long as the bandwidth resources in the WDM network are sufficient.

Moreover, we assume that the DC backup process makes all the DCs conduct their backups simultaneously during a period when the network is relatively unoccupied (e.g., at midnights). Hence, DC-B-Wnd becomes the time duration to back up all the newly-generated data in the DCs. In order to optimize the backup process, we assume that the inter-DC network operates as a discrete-time system [23], [24], and the backup scenario (i.e., the backup pairs and the corresponding data-transfer paths) can be reconfigured every time interval Δt. Note that the choice of Δt really depends on the NC&M mechanism in the inter-DC network. Considering the fact that it may take the operator more than a minute to reconfigure a lightpath [24], [29], we expect Δt to be on the magnitude of tens of minutes, which is similar to the assumptions used in other studies on traffic scheduling in optical networks [30]–[32]. Moreover, there is a tradeoff between DC-B-Wnd and operational complexity, which is controlled by Δt. Specifically, a smaller Δt means that the network is optimized more frequently (i.e., higher operational complexity) and hence may lead to a shorter DC-B-Wnd, and *vice versa*. We will provide detailed analysis on this tradeoff in Section VI. Within each Δt, we use the "1-on-1" scheme. That is to say, as a production DC, it only selects one remote DC as its backup site, and as a backup DC, it can only receive data from one production DC as well.

The physical topology is modeled as a directed graph $G(V, E)$, where V and E represent the sets of nodes and fiber links, respectively. Note that V includes two types of nodes: 1) the DC nodes that each has a local DC and an OXC, and 2) the switch nodes that do not have local DCs. We denote the set of DC nodes as $V^{\mathrm{dc}} \subseteq V$, where $|V^{\mathrm{dc}}| = K$. Each link $(v, u) \in E$ has a bandwidth capacity $B_{(v,u)}$, which is in the number of wavelength channels. A_v denotes the amount of data to be backed up in DC node $v \in V^{\mathrm{dc}}$, and the total data-transfer in the backup is $M = \sum_{v \in V^{\mathrm{dc}}} A_v$.

In this work, we define the disaster zone (DZ) as the set of DC nodes that might be impacted simultaneously by a single natural disaster. Hence, in order to prevent the incident that the data can never be restored, we ensure that in each backup pair, the production DC and its backup site do not fall into the same DZ. For a DC node v, its DZ is denoted as V_v^z. Note that since we use the mutual backup model and assume that the DCs can serve as the backup sites of one another, we ignore the issues on cloud security and data privacy. However, in real geo-distributed DCs, one needs to consider the data protection laws since the data transfers among DCs should obey them [33]. Fortunately, our model can be easily extended to consider these issues. Basically, we can extend the definition of DZ and make the DZ become the forbidden zone, which includes not only the DCs that might be impacted simultaneously by a single disaster but also those that should not share data between each other according to the data protection laws. Table I summarizes the important notations that are used in this work.

IV. ILP Formulation

In this section, we formulate an ILP model based on the network model presented in the previous section, and use it to optimize the DC backup scenario in the optical inter-DC network, with the objective to minimize DC-B-Wnd. Basically, the model takes network status as the input, and optimizes the backup scenario (i.e., the backup pair and data-transfer paths) in every time interval Δt for each production DC. Here, we leverage the "anycast" communication scheme [26], [34], [35] for the DC backup. Specifically, the backup site of a DC is selected

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.2015.2425303, Journal of Lightwave Technology

TABLE I
NOTATIONS USED IN PROBLEM FORMULATION

Notations	Definitions
$G(V, E)$	Physical topology of the network
$B_{(v,u)}$	Capacity of link (v, u) in wavelength channels
K	Number of DC nodes
V^{dc}	Set of DC nodes
V_v^z	DZ of DC v
A_v	Amount of data to be backed up on DC v
M	Total amount of data to be backed up on all the DCs
Δt	Time interval for network operation
N^{ub}	Upper-bound of the number of time intervals used for the DC backup
$\mathbb{M}$	A large integer that satisfies $\mathbb{M} \geq \sum_{(v,u)\in E} B_{(v,u)}$
T	DC-B-Wnd

adaptively according to the network status, and can be changed in between adjacent intervals. As the service for DC backup can log the actual backup scenario in each interval [23], changing the backup site of a DC will not cause confusion. Meanwhile, the changes on the data-transfer paths can be realized by using the existing lightpath reconfiguration techniques, e.g., those in [28], [29].

First of all, with the bottleneck link in the network, we estimate the upper-bound of the number of time intervals for finishing the backup process as

$$N^{ub} = \lceil \frac{M}{\min\limits_{(v,u)\in E}(B_{(u,v)}) \cdot \Delta t} \rceil \tag{1}$$

and then the upper-bound of DC-B-Wnd is $N^{ub} \cdot \Delta t$.

Then, the ILP formulation is as follows.

Notations:

1) $G(V, E)$: Physical topology of the network.
2) $B_{(v,u)}$: Capacity of link (v, u) in wavelength channels.
3) K: Number of DC nodes.
4) V^{dc}: Set of DC nodes.
5) V_v^z: DZ of DC v.
6) A_v: Amount of data to be backed up on DC v.
7) M: Total amount of data to be backed up on all the DCs.
8) Δt: Time interval.
9) N^{ub}: Upper-bound of the number of time intervals used for the DC backup.
10) $\mathbb{M}$: A large integer that satisfies $\mathbb{M} \geq \sum_{(v,u)\in E} B_{(v,u)}$.

Variables:

1) T: Integer variable that represents DC-B-Wnd.
2) $x_{\{v,u\}}^{(j)}$: Boolean variable that equals 1 if DC v uses the backup pair $\{v, u\}$ in the jth time interval, and 0 otherwise. Note that if v is not a DC node we have $x_{\{v,u\}}^{(j)} = 0$, i.e., $x_{\{v,u\}}^{(j)} = 0,\ \forall v, u \in V \setminus V^{\mathrm{dc}}, \forall j \in [1, N^{ub}]$.
3) $\pi_{\{v,u\},(w,z)}^{(j)}$: Integer variable that indicates the number of wavelength channels allocated on link (w, z) for the backup pair $\{v, u\}$ in the jth time interval. Here, $\pi_{\{v,u\},(w,z)}^{(j)} = 0\ \forall v, u \in V \setminus V^{\mathrm{dc}}, \quad \forall j \in [1, N^{ub}]$.
4) $d_{\{v,u\}}^{(j)}$: Integer variable that indicates the total number of wavelength channels used by the backup pair $\{v, u\}$ in the jth time interval. Again, $d_{\{v,u\}}^{(j)} = 0\ \forall v, u \in V \setminus V^{\mathrm{dc}}, \quad \forall j \in [1, N^{ub}]$.
5) N_v: Integer variable that indicates the number of time intervals used for finishing the backup on DC node v.

Objective: The objective of the ILP is to minimize DC-B-Wnd T,

$$\text{Minimize} \quad T = \Delta t \cdot \max_{v\in V^{\mathrm{dc}}}(N_v). \tag{2}$$

In order to make the objective linear, we modify Eq. (2) to

$$T \geq \Delta t \cdot N_v, \quad \forall v \in V^{\mathrm{dc}}. \tag{3}$$

Constraints:

$$x_{\{v,v\}}^{(j)} = 0, \quad \forall v \in V^{\mathrm{dc}}, \quad j \in [1, N^{ub}] \tag{4}$$

$$\sum_{u\in V^{\mathrm{dc}}} x_{\{v,u\}}^{(j)} \leq 1, \quad \forall v \in V^{\mathrm{dc}}, \quad j \in [1, N^{ub}] \tag{5}$$

$$\sum_{v\in V^{\mathrm{dc}}} x_{\{v,u\}}^{(j)} \leq 1, \quad \forall u \in V^{\mathrm{dc}}, \quad j \in [1, N^{ub}]. \tag{6}$$

Eqs. (4)–(6) are for the constraints from the "1-on-1" backup scheme. Eq. (4) ensures that a production DC will not select itself as the backup site, while Eqs. (5)–(6) make sure that a production DC only has one backup site and a DC only receives backup data from one production DC in each interval

$$x_{\{v,u\}}^{(j)} = 0, \quad \forall v \in V^{\mathrm{dc}}, u \in V_v^z, j \in [1, N^{ub}]. \tag{7}$$

Eq. (7) ensures that each backup pair will not fall into the same DZ

$$\sum_{w:(w,z)\in E} \pi_{\{v,u\},(w,z)}^{(j)} - \sum_{w:(z,w)\in E} \pi_{\{v,u\},(z,w)}^{(j)} = \begin{cases} -d_{\{v,u\}}^{(j)}, & z = v, \\ d_{\{v,u\}}^{(j)}, & z = u \\ 0, & \text{Otherwise.} \end{cases} \quad \forall v \in V^{dc}, j \in [1, N^{ub}]. \tag{8}$$

Eq. (8) enforces the flow conservation for the data-transfer of each backup pair $\{v, u\}$, which means that at any node $z \in V$, the outgoing and incoming lightpaths should be equal, except for the source and destination, i.e., v and u

$$d_{\{v,u\}}^{(j)} \leq x_{\{v,u\}}^{j} \cdot \mathbb{M}, \quad v \in V^{\mathrm{dc}}, j \in [1, N^{ub}]. \tag{9}$$

Eq. (9) ensures that no wavelength channels are allocated to the lightpath $v \to u$, if (v, u) is not a backup pair

$$\sum_{v\in V^{\mathrm{dc}}} \sum_{u\in V^{\mathrm{dc}}} \pi_{\{v,u\},(w,z)}^{(j)} \leq B_{(w,z)}\ \forall (w, z) \in E, j \in [1, N^{ub}]. \tag{10}$$

Eq. (10) ensures that the number of allocated wavelength channels on each link will not exceed its bandwidth capacity

$$N_v \geq j \cdot x_{\{v,u\}}^{(j)} \quad \forall v, u \in V_{\mathrm{dc}}, j \in [1, N^{ub}]. \tag{11}$$

Eq. (11) obtains the value of N_v for each production DC v

$$\Delta t \cdot \sum_{j=1}^{N^{ub}} \sum_{u\in V^{\mathrm{dc}}} d_{\{v,u\}}^{(j)} \geq A_v \quad \forall v \in V^{\mathrm{dc}}. \tag{12}$$

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.2015.2425303, Journal of Lightwave Technology

Algorithm 1: Overall FR-based Algorithm

input : $G(V,E), V^{dc}, V_v^z, A_v$
output: DC-B-Wnd T

1 $T=0, R^{dc}=V^{dc}, j=1$;
2 **while** $R^{dc}\neq\emptyset$ **do**
3 determine the current Backup Scenario and the bandwidth assignments $\{D_v, \forall v\in V^{dc}\}$;
4 $T=T+\Delta t, j=j+1$;
5 **for** all $v\in R^{dc}$ **do**
6 $A_v=A_v-D_v\cdot\Delta t$;
7 **if** $A_v\leq 0$ **then**
8 $R^{dc}=R^{dc}\setminus v$;
9 **end**
10 **end**
11 **end**

Eq. (12) ensures that the backup process transfers all the data.

The numbers of variables and constraints used in this ILP model depend on N^{ub} and $G(V,E)$, and N^{ub} is determined by the total amount of data to be backed up M and the interval Δt. Therefore, the ILP can become intractable with high computational complexity for a large-scale problem that has a relatively large network topology, and/or huge amounts of data to be backed up, and/or a short interval Δt.

V. Heuristic Algorithms

The ILP model can optimize the DC backup process in the optical inter-DC network by minimizing DC-B-Wnd and providing the exact solutions. However, as it considers the selection of backup sites and data-transfer paths jointly, the ILP becomes intractable and does not scale well for large problems. In order to improve the time efficiency, we propose several heuristics in this section.

A. Algorithms Based on Fixed Reconfiguration (FR)

In order to reduce the computational complexity, we leverage a greedy idea that in a Δt, if there is available bandwidth and its data to be backed up has not been fully transferred yet, a production DC will establish the lightpaths for data-transfer immediately, instead of waiting for the next Δt to explore better backup scenario design. The overall procedure of the proposed heuristic is illustrated in *Algorithm* 1. *Line* 1 is for the initialization, and here we define R^{dc} as the set of DC nodes whose data-transfers have not been finished yet.

Definition We define the *Backup Scenario* as the arrangement of the backup pairs and the related data-transfer paths in the inter-DC network within a certain time interval Δt.

In the while-loop in *Lines* 2–11, we obtain the Backup Scenario within each interval, until all the DC $v\in V^{\mathrm{dc}}$ have accomplished the data backup. Note that here, we can use different sub-routines in *Line* 3 to determine the Backup Scenario and get the bandwidth assigned to DC v (i.e., D_v, in the number of wavelength channels). The details on the sub-routines will be discussed in Section V-C. Since *Algorithm* 1 tries to reconfigure the Backup Scenario every time interval Δt, we refer to it as a FR based one.

Algorithm 2: Overall AR-based Algorithm

input : $G(V,E), V^{dc}, V_v^z, A_v$
output: DC-B-Wnd T

1 $T=0, R^{dc}=V^{dc}, j=1$;
2 **while** $R^{dc}\neq\emptyset$ **do**
3 sort $\{A_v\}$ in descending order to get $Rank^{(j)}$;
4 **if** $Rank^{(j)}\neq Rank^{(j-1)}$ **then**
5 determine the current Backup Scenario and the bandwidth assignments $\{D_v, \forall v\in V^{dc}\}$;
6 **else**
7 keep the Backup Scenario unchanged;
8 **end**
9 $T=T+\Delta t, j=j+1$;
10 **for** all $v\in R^{dc}$ **do**
11 $A_v=A_v-D_v\cdot\Delta t$;
12 **if** $A_v\leq 0$ **then**
13 $R^{dc}=R^{dc}\setminus v$;
14 **end**
15 **end**
16 **end**

B. Algorithms Based on AR

In the FR-based algorithm, the Backup Scenario is re-optimized every Δt. However, this may result in increased operational complexity. Therefore, we consider an AR approach in this subsection, which can balance the tradeoff between the performance on DC-B-Wnd and the operational complexity. Specifically, it checks whether it is necessary to change the Backup Scenario at the beginning of each Δt. *Algorithm* 2 shows the detailed procedure. In *Line* 3, we sort the remaining data in all the DCs $\{A_v\}$ in descending order, and get the set of ranks for the j-th interval as $Rank^{(j)}$. *Line* 4 checks whether the ranks are the same as that in the previous interval. If yes, we keep the Backup Scenario unchanged. Otherwise, we re-optimize the Backup Scenario with a sub-routine. The rationale behind this is that since the DC that has the most data to be backed up usually contributes the most to DC-B-Wnd, we should re-optimize the Backup Scenario if the ranks change. The rest of the operations is the same as those in *Algorithm* 1. As *Algorithm* 2 reconfigures the Backup Scenario adaptively according to the network status, we refer to it as an AR based one.

C. Sub-routines to Determine Backup Scenario

In this subsection, we propose several heuristic algorithms for the sub-routine to determine the Backup Scenario for a specific interval Δt based on the network status.

1) OneStep-GMF Algorithm: We first propose a heuristic based on the global maximum flow (GMF). Here, the idea is to preferentially allocate bandwidth resources to the backup pair that has the largest flow throughput. Algorithm 3 shows the

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.2015.2425303, Journal of Lightwave Technology

Algorithm 3: OneStep-GMF Algorithm

input : $G(V,E), V^{dc}, V_v^z, A_v, R^{dc}$
output: Total wavelength usage D_v

1 $R_{(j)}^{dc} = R^{dc}, V_{(j)}^{dc} = V^{dc}, D_v = 0, \forall v \in R^{dc}$;
2 **while** $R_{(j)}^{dc} \neq \emptyset$ **do**
3 **for** each $v \in R_{(j)}^{dc}$ **do**
4 **for** each $u \in V_{(j)}^{dc} \setminus V_v^z$ **do**
5 compute the MF $f_{\{v,u\}}$ for $v \rightarrow u$;
6 **end**
7 **end**
8 get the GMF in $\{f_{\{v,u\}}\}$;
9 **if** $f_{\{v,u\}} > 0$ **then**
10 $D_v = f_{\{v,u\}}$;
11 set data-transfer with $f_{\{v,u\}}$ for $v \rightarrow u$;
12 $R_{(j)}^{dc} = R_{(j)}^{dc} \setminus v, V_{(j)}^{dc} = V_{(j)}^{dc} \setminus u$;
13 update network status;
14 **else**
15 **break**;
16 **end**
17 **end**

Algorithm 4: OneStep-MDF Algorithm

input : $G(V,E), V^{dc}, V_v^z, A_v, R^{dc}$
output: Total wavelength usage D_v

1 $R_{(j)}^{dc} = R^{dc}, V_{(j)}^{dc} = V^{dc}, D_v = 0, \forall v \in R^{dc}$;
2 **for** each $v \in R_{(j)}^{dc}$ in descending order of A_v **do**
3 **for** each $u \in V_{(j)}^{dc} \setminus V_v^z$ **do**
4 compute the MF $f_{\{v,u\}}$ for $v \rightarrow u$;
5 **end**
6 get the largest $f_{\{v,u\}}$ in $\{f_{\{v,u\}}\}$;
7 **if** $f_{\{v,u\}} > 0$ **then**
8 $D_v = f_{\{v,u\}}$;
9 set data-transfer with $f_{\{v,u\}}$ for $v \rightarrow u$;
10 $V_{(j)}^{dc} = V_{(j)}^{dc} \setminus u$;
11 update network status;
12 **else**
13 **break**;
14 **end**
15 **end**

detailed procedure. In Lines 3–7, we calculate the maximum flows (MFs) for all the feasible backup pairs in $G(V,E)$, and then Line 8 selects the MF that has the largest throughput (i.e., GMF) to set up the data-transfer for the corresponding backup pair. For an interval Δt, the loop that covers Lines 2–17 repeats the operations above until all the DCs are served or the bandwidth resources are used up. For each DC, Algorithm 3 gets the backup pair and data-transfer paths in one step, and hence can be named as an one-step algorithm (*OneStep-GMF*).

The while-loop from Line 2 to 17 can execute $|V^{\mathrm{dc}}|$ times, for selecting the backup pairs. For checking all the feasible MFs, the for-loop from Line 3 to 7 runs $|V^{\mathrm{dc}}|^2$ times, while in Line 5, the time complexity for calculating a MF is $\mathcal{O}(|V| \cdot |E|^2)$ [36]. Hence, the overall time complexity of *OneStep-GMF* is $\mathcal{O}(|V^{\mathrm{dc}}|^3 \cdot |V| \cdot |E|^2)$.

2) OneStep-MDF Algorithm: As Algorithm 3 needs to calculate all the feasible MFs to find a backup pair, its time complexity is still high. Meanwhile, as the production DC that has the most data to be backed up usually contributes the most to DC-B-Wnd, serving such a DC with the highest priority should help us reduce DC-B-Wnd. Algorithm 4 shows a heuristic that handles the DC that has the most data to be backed up first and chooses the backup site that can provide the largest data-transfer throughput for it (*OneStep-MDF*). *Line* 2 sorts the DCs in descending order of the remaining data to be backed up. Then, in Lines 3–5, we calculate the MFs from a DC to all the feasible backup sites, and select the one that provides the largest flow throughput.

The for-loop from Line 2 to 15 runs $|V^{\mathrm{dc}}|$ times. Similar to *OneStep-GMF*, the for loop from *Line* 3–5 has a time complexity of $\mathcal{O}(|V^{\mathrm{dc}}| \cdot |V| \cdot |E|^2)$. Then, the overall time complexity of *OneStep-MDF* is $\mathcal{O}(|V^{\mathrm{dc}}|^2 \cdot |V| \cdot |E|^2)$.

3) TwoStep-MDF Algorithm: Algorithms 3 and 4 both select the backup site and data-transfer paths in one step, and do not consider the geographic distance between the backup pairs. Note that if the geographic distance between a backup pair is relatively long, the corresponding data-transfer can consume bandwidth resources on a large number of fiber links and hence limit the throughput of other backup pairs. Consequently, we may have a prolonged DC-B-Wnd. Algorithm 5 tries to address this issue by adopting a two-step approach, i.e., obtaining the backup site and data-transfer paths separately. In this algorithm, we still sort the DCs in descending order of remaining data to be backed up, and then process them one by one, as shown in Lines 2–13. But in Line 3, a DC's backup site is chosen such that the shortest routing path between the backup pair has the smallest hop-count. Then, Line 4 calculates the MF and uses it to set up the data-transfer for the backup pair. We denote Algorithm 5 as *TwoStep-MDF*.

For this algorithm, the for-loop from Line 2 to Line 13 runs $|V^{\mathrm{dc}}|$ times. Hence, the time complexity of *TwoStep-MDF* is $\mathcal{O}(|V^{\mathrm{dc}}| \cdot |V| \cdot |E|^2)$.

4) TwoStep-ILP Algorithm: Finally, we discuss an algorithm that can optimize the data-transfers for all the backup pairs simultaneously. Algorithm 6 shows the detailed procedure, which still uses the two-step approach, i.e., 1) selecting the backup sites for all the DCs based on the geographic distances, and 2) determining the data-transfer paths with the joint consideration of the bandwidth resources and data to be backed up. For each of the two steps, we formulate an ILP to solve the problem and the details are discussed as follows.

We formulate the first ILP (*Step1-ILP*) to determine the backup pairs for all the DCs simultaneously, with the objective to minimize the summation of the backup pairs' hop-counts.

Notations:

1) $H_{\{v,u\}}$: Hop-count of the shortest path between the backup pair $\{v,u\}$.

Variables:

1) $x_{\{v,u\}}$: Boolean variable that equals 1 if DC v uses the backup pair $\{v,u\}$ in the current time interval, and 0 otherwise. Note that if v is not a DC node we have $x_{\{v,u\}} = 0$, i.e., $x_{\{v,u\}} = 0,\ \forall v,u \in V \setminus V^{\mathrm{dc}}$.

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.2015.2425303, Journal of Lightwave Technology

Algorithm 5: TwoStep-MDF Algorithm

input : $G(V,E), V^{dc}, V_v^z, A_v, R^{dc}$
output: Total wavelength usage D_v

$R_{(j)}^{dc} = R^{dc}, V_{(j)}^{dc} = V^{dc}, D_v = 0, \forall v \in R^{dc}$;
for each $v \in R_{(j)}^{dc}$ in descending order of A_v **do**
  get u from $V_{(j)}^{dc} \setminus V_v^z$, whose hop-count to v is the smallest;
  compute the MF $f_{\{v,u\}}$ for $v \to u$;
  if $f_{\{v,u\}} > 0$ **then**
    $D_v = f_{\{v,u\}}$;
    set data-transfer with $f_{\{v,u\}}$ for $v \to u$;
    $V_{(j)}^{dc} = V_{(j)}^{dc} \setminus u$;
    update network status;
  else
    break;
  end
end

Objective:

$$\text{Minimize} \quad \sum_{v \in V^{\mathrm{dc}}} \sum_{u \in V^{\mathrm{dc}}} (H_{\{v,u\}} \cdot x_{\{v,u\}}). \tag{13}$$

Constraints:

$$x_{\{v,v\}} = 0, \quad \forall v \in V^{\mathrm{dc}}, \tag{14}$$

$$\sum_{u \in V^{\mathrm{dc}}} x_{\{v,u\}} = 1, \quad \forall v \in V^{\mathrm{dc}}, \tag{15}$$

$$\sum_{v \in V^{\mathrm{dc}}} x_{\{v,u\}} = 1, \quad \forall u \in V^{\mathrm{dc}}, \tag{16}$$

$$x_{\{v,u\}} = 0, \quad \forall v \in V^{\mathrm{dc}}, u \in V_v^z. \tag{17}$$

Eqs. (14)–(17) are for the constraints on backup site selection, which are similar to those discussed in Section IV.

Proposition 1. The problem described by *Step1-ILP* can be solved in polynomial time.

Proof. We first build an auxiliary graph that contains two columns of nodes, based on the topology of the inter-DC network $G(V,E)$. Each node in the auxiliary graph represents a DC node $v \in V^{\mathrm{dc}}$, while the left column includes all the DCs and those in the right column are their potential backup sites. Since we use the mutual backup model, both columns include all the DC nodes. Here, we define the sets of nodes in the two columns as X and Y, where $X = Y = V^{\mathrm{dc}}$. For any two nodes $x \in X$ and $y \in Y$ in the auxiliary graph, there is an edge to connect them *if and only if* y can be selected as the backup site for x, i.e., $y \notin V_x^z$, and the edge's value is $-H_{\{x,y\}}$, where $H_{\{x,y\}}$ denotes the hop-count of the shortest path between x and y in $G(V,E)$. With this auxiliary graph, we can transform the problem described in the ILP into a maximum weighted bipartite graph matching problem [37], which tries to find the 1-to-1 matching among the nodes in the two columns such that the total value of the selected edges is maximized. Since the Hungarian Algorithm [36] can solve the aforementioned bipartite graph matching problem with a time complexity of $\mathcal{O}(|V^{\mathrm{dc}}|^4)$ for the worst case, we prove that *Step1-ILP* can be solved in polynomial time.

Algorithm 6: TwoStep-ILP Algorithm

input : $G(V,E), V^{dc}, V_v^z, A_v, R^{dc}$
output: Total wavelength usage D_v

obtain backup pairs using *Step1-ILP*;
compute MFs $\{f_{\{v,u\}}\}$ using *Step2-ILP*;
$D_v = 0, \forall v \in R^{dc}$;
for *each* $v \in R^{dc}$ **do**
  get MF $f_{\{v,u\}}$ in $\{f_{\{v,u\}}\}$;
  if $f_{\{v,u\}} > 0$ **then**
    $D_v = f_{\{v,u\}}$;
    set data-transfer with $f_{\{v,u\}}$ for $v \to u$;
    update network status;
  end
end

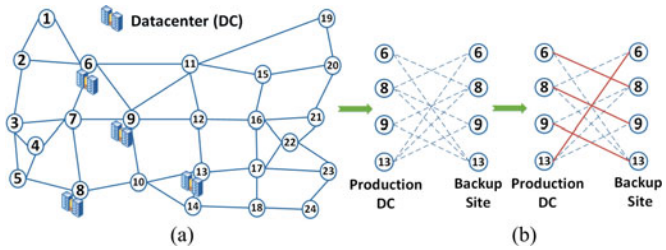

Fig. 2. Example of using an auxiliary graph and bipartite graph matching to determine the backup pairs.

Fig. 2 shows an example of the construction of the auxiliary graph and the consequent weighted bipartite graph matching. It can be seen that the nodes in the two columns are connected according to the feasible backup pairs. Here, we assume that the DZ V_v^z covers all the nodes that are within one hop from v, e.g., Node 9 cannot select Node 6 as its backup site. Therefore, Nodes 6 and 9 are not connected in the auxiliary graph in Fig. 2(b). Finally, the bipartite graph matching gets the backup pairs as those connected with red solid lines in Fig. 2(c).

Then, in order to get the data-transfer paths for the backup pairs determined in *Step1-ILP*, we formulate the second ILP (*Step2-ILP*) to maximize the summation of the products of the remaining data and the backup throughput for all the DCs. The rationale behind using the products is that the optimization should consider the remaining data and the backup throughput jointly, as they both affect DC-B-Wnd.

Notations:

1) A_v: Remaining data to be backed up on v before the current time interval.
2) P: Set of backup pairs $\{\{v,u\}\}$ got in the previous step.

Variables:

1) $\pi_{\{v,u\},(w,z)}$: Integer variable that indicates the number of allocated wavelength channels on link (w,z) for the backup pair $\{v,u\}$.
2) $d_{\{v,u\}}$: Integer variable that indicates the total allocated wavelength channels for the backup pair $\{v,u\}$.

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.2015.2425303, Journal of Lightwave Technology

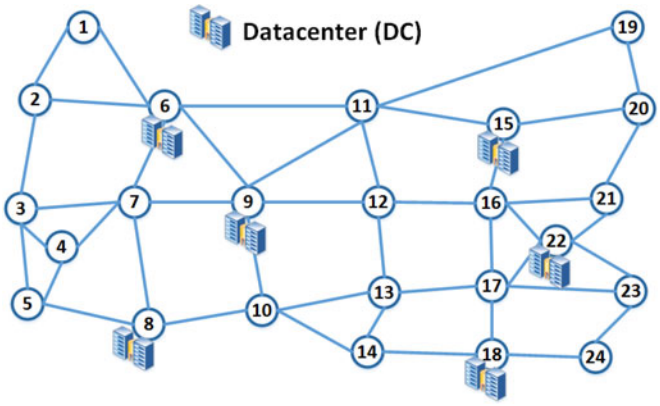

Fig. 3. Optical inter-DC network with US-Backbone topology.

Objective:

$$\text{Maximize} \quad \sum_{\{v,u\}\in P} (A_v \cdot d_{\{v,u\}}). \tag{18}$$

Constraints:

$$\sum_{w:(w,z)\in E} \pi_{\{v,u\},(w,z)} - \sum_{w:(z,w)\in E} \pi_{\{v,u\},(z,w)} = \begin{cases} -d_{\{v,u\}}, & z = v, \\ d_{\{v,u\}}, & z = u \\ 0, & \text{Otherwise.} \end{cases} \quad \forall v \in V^{dc}, \tag{19}$$

$$\sum_{(v,u)\in P} \pi_{\{v,u\},(w,z)} \leq B_{(w,z)} \ \forall (w,z) \in E. \tag{20}$$

Eqs. (19)–(20) are for the constraints on flow conservation and bandwidth limitation, which are also similar to those discussed in Section IV.

In order to solve *Step2-ILP* in polynomial time, we convert it into a *Relaxed-LP* model in which we relax both $\pi_{\{v,u\},(w,z)}$ and $d_{\{v,u\}}$ to non-negative float values. Then, we design a *Rounding Algorithm* to obtain the integral solution. Specifically, the *Rounding Algorithm* makes sure that the capacity constraint and flow conservation constraint are satisfied. For the capacity constraint, we round down the float capacity solution obtained by the *Relaxed-LP*, and obtain the rounded integral capacity solution as a set $\mathcal{F}_v$, in which each element indicates the wavelength usage on a link $e \in E$ for DC v. Note that the rounding may cause violations to the flow conservation constraint. To address this issue, we construct an auxiliary graph $G_v^a(V_v^a, E_v^a)$ for DC v, in which we have $V_v^a = V$ and $E_v^a = \mathcal{F}_v$. Then, we calculate the MF in $G_v^a(V_v^a, E_v^a)$ from v to u that satisfies $\{v,u\} \in P$, and finally obtain the integral capacity solution for DC v.

Note that either the ellipsoid algorithm or the interior point algorithm can solve the *Relaxed-LP* in polynomial time [38]. As the *Relaxed-LP* is similar to the multi-commodity flow problem, it can be solved in $\mathcal{O}(|E|^{3.5})$ [39]. Then, the time complexity of *Step2-ILP* is $\mathcal{O}(|E|^{3.5} + |V^{\mathrm{dc}}| \cdot |V| \cdot |E|^2)$. In all, *TwoStep-ILP* can be solved in polynomial time and its overall time complexity is $\mathcal{O}(|V^{\mathrm{dc}}|^4 + |E|^{3.5} + |V^{\mathrm{dc}}| \cdot |V| \cdot |E|^2)$.

VI. Performance Evaluation

In this section, we present performance evaluations by solving the ILP model in Section IV and simulating the heuristics proposed in the previous section. All the simulations use the US-Backbone topology shown in Fig. 3, which includes six geo-distributed DC nodes, as Nodes 6, 8, 9, 15, 18 and 22. We assume that for a DC node v, its DZ V_v^z covers all the nodes that are within one-hop from it. Before the DC backup process, the available capacity of each link $B_{(v,u)}$ is uniformly distributed within $[10, 30]$ wavelength channels. This is because the backup process happens in an operational inter-DC network that may carry other live traffic too.

In the simulations, we normalize Δt and M in time-units and units, respectively, for obtaining the general trend of the proposed algorithms' performance, but do not use specific numbers. Similarly, the throughput of lightpaths is quantified with the number of wavelength channels, but not in Gb/s. Note that even though the specific numbers are not used for these parameters, we do make sure that the relation among them matches with the real case and is reasonable. For instance, we have $\Delta t = 30$ min, set the data-rate of a wavelength channel as 40 Gb/s, make each fiber link contain $[10, 30]$ available wavelength channels. Then, according to the simulation results in this section, the normal value of DC-B-Wnd falls in $[1, 5]$ time-units. For a simple estimation, the maximum amount of data to be backed up would be $5 \times 30 \times 60 \times 40 \times 30/8 = 1350000$ GBytes $= 1.35$ PBytes. So the data to be backed up is in PBytes. Meanwhile, it is known that large enterprises like Google can process 100 PBytes data daily in its DCs [40]. Normally, with incremental backup and data compression, it needs to back up $< 5\%$ of the production data [41]. Hence, we can see that the results obtained with our parameters match with the real case for Google.

In each set of simulations, we keep $M = \sum_{v\in V^{\mathrm{dc}}} A_v$ as fixed but randomly change A_v on each DC v. To obtain each data point, we carry out 100 tests and then average out the results. By doing so, we make sure that the results have sufficient statistical accuracy. The simulation environment is a Windows server that has an Intel-Xeon 2.40 GHz CPU and 32 GB memory. According to Section V, the heuristic can be either an FR- or AR-based one for the overall procedure, and then for the sub-routine to determine the Backup Scenario, we can choose one from *OneStep-GMF*, *OneStep-MDF*, *TwoStep-MDF* and *TwoStep-ILP*. Hence, there are 8 heuristics to evaluate and we name them with the combination of the overall procedure and sub-routine. For instance, for the FR-based one that uses *OneStep-GMF*, we denote it as *FR-OneStep-GMF*.

Since compared with the AR-based ones, the FR-based algorithms optimize the Backup Scenario more frequently, they may get DC-B-Wnds that are closer to the optimal solution from the ILP model in Section IV. The simulations first compare the DC-B-Wnds from the ILP and FR-based algorithms, and analyze the impact of the total amount of data to be backed up (i.e., M) on DC-B-Wnd. We also compare the computation time of the ILP and FR-based algorithms. Then, we compare the FR-based algorithms with the AR-based ones in terms of DC-B-Wnd and reconfiguration times, to explore the tradeoff between

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.2015.2425303, Journal of Lightwave Technology

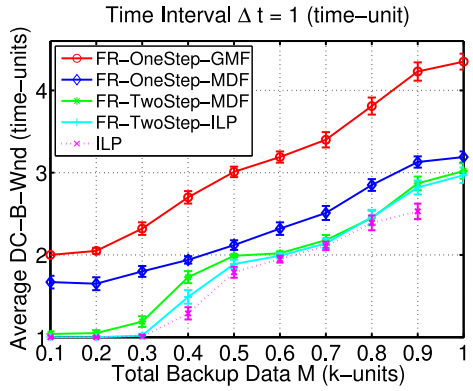

Fig. 4. Results on DC-B-Wnd versus total amounts of data M.

DC-B-Wnd and operational complexity. In order to ensure sufficient statistic validity, we also plot 90% confidence intervals of the results.

A. Comparisons of FR-Based Heuristics and ILP

1) Results on DC-B-Wnd: Fig. 4 shows the results on DC-B-Wnd when M changes. Here, we set the time interval $\Delta t = 1$ time-unit. The results indicate among all the FR-based heuristics, *FR-TwoStep-ILP* provides the results on DC-B-Wnd that are closest to the optimal ones obtained by the ILP, while *FR-OneStep-GMF* gives the longest DC-B-Wnds. We also observe that when M is larger than 0.9 k-units, solving the ILP becomes too time-consuming and hence we cannot get the optimal solution. Note that DC-B-Wnd depends on the bandwidth resources in the network, the amount of data to be backed up on each DC, and the selection of the backup pairs. Among *FR-OneStep-GMF*, *FR-OneStep-MDF* and *FR-TwoStep-MDF*, since *FR-OneStep-GMF* only considers the bandwidth resources, it provides the worst results on DC-B-Wnd. *FR-OneStep-MDF* optimizes the backup process with joint consideration of both the bandwidth resources and the amount of backup data on each DC, and hence it outperforms *FR-OneStep-GMF*. On top of these two, *FR-TwoStep-MDF* takes care of all the three factors and thus achieves the shortest DC-B-Wnds among the three. However, as these three algorithms optimize the data-transfers for the backup pairs one-by-one in the greedy manner, they may only maximize the data-transfer for one backup pair and limit the bandwidth for others. *FR-TwoStep-ILP* overcomes this issue by considering all the data-transfers jointly, and that is why it achieves the shortest DC-B-Wnds among all the FR-based heuristics.

Table II shows the impacts of Δt on DC-B-Wnd, with $M = 1200$ units. It can be seen that for the algorithms, the results on DC-B-Wnd follow the similar trends as those in Fig. 4. DC-B-Wnd generally increases with Δt, but it is also interesting to notice that for certain cases, a larger Δt may result in a smaller DC-B-Wnd. Intuitively, a larger Δt means less frequent network optimizations and hence leads to a longer DC-B-Wnd. However, as Δt is an integer, a data-transfer may only occupy a portion of it upon finishing. The ceiling operation is the reason why DC-B-Wnd can decrease with Δt. For instance, for $\Delta t = 4$ time-units, even if the actual DC-B-Wnd is 12.1 time-units, the ceiling result is 16 time-units. While for $\Delta t = 5$ time-units, an actual DC-B-Wnd of 14.9 time-units leads to the final result as 15 time-units.

TABLE II
AVERAGE DC-B-WND IN TIME-UNITS FOR $M = 1200$ UNITS

Algorithms	Time Interval Δt (time-units)					
	4	5	6	7	8	9
FR-OneStep-GMF	12.92	14.9	16.68	17.29	18.8	20.07
FR-OneStep-MDF	9.08	10.4	11.88	12.88	15.04	15.84
FR-TwoStep-MDF	8.24	9.65	10.38	10.36	9.6	9.9
FR-TwoStep-ILP	8.24	9.3	9.06	8.26	8.56	9.18
ILP	8.08	8.7	8.04	7.49	8.4	9.18

TABLE III
AVERAGE COMPUTATION TIME IN SECONDS FOR $\Delta t = 1$ TIME-UNIT

Algorithms	Total Backup Data M (k-units)				
	0.6	0.7	0.8	0.9	1
FR-OneStep-GMF	5.5682	6.0714	6.6238	7.1104	7.3882
FR-OneStep-MDF	1.5506	1.6601	1.905	2.0242	2.1518
FR-TwoStep-MDF	0.4362	0.4696	0.5138	0.5666	0.6237
FR-TwoStep-ILP	0.467	0.5201	0.5546	0.6162	0.6698
ILP	18.696	23.379	33.215	50.49	N/A

2) Results on Computation Time: Table III presents the results on the computation time when we use $\Delta t = 1$ time-unit. We observe that when $M = 1000$ units, the ILP cannot obtain the optimal result due to the high time complexity. Basically, when the problem size becomes larger, the computation time of the ILP increases exponentially, but the computation time of the heuristics increases with a more moderate slope. The results on the computation time also verify our analysis on the algorithms' time complexities in Section V. It can be seen that among the heuristics, *FR-OneStep-GMF* consumes the longest computation time since it enumerates all the DCs and the corresponding feasible backup sites to find a backup pair. *FR-OneStep-GMF* is followed by *FR-OneStep-MDF* and *FR-TwoStep-MDF*, which can only enumerate all the backup sites or use the pre-computed backup sites, respectively. *FR-TwoStep-ILP* also uses pre-computed backup sites as well and can take advantage of the ellipsoid or interior point algorithm to solve the *Relaxed-LP* with fast speed, but the subsequent *Rounding Algorithm* in it increases the time complexity. This algorithm takes slightly longer time than *FR-TwoStep-MDF*, and is the second fastest algorithm.

In all, the simulations discussed above indicate that *FR-TwoStep-ILP* is an efficient and promising algorithm, since among the four FR-based heuristics, it obtains the shortest DC-B-Wnds with a relatively short computation time.

B. Comparisons of FR- and AR-Based Heuristics

We then present simulation results on DC-B-Wnd, reconfiguration times and computation time, and use them to compare the performance of FR- and AR-based heuristics.

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.2015.2425303, Journal of Lightwave Technology

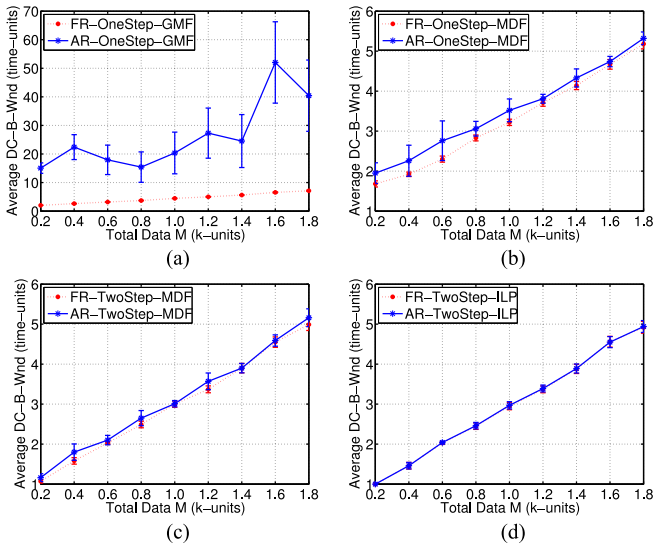

Fig. 5. Comparison on DC-B-Wnds from *AR-based* and *FR-based* heuristics when $\Delta t = 1$ (time-unit).

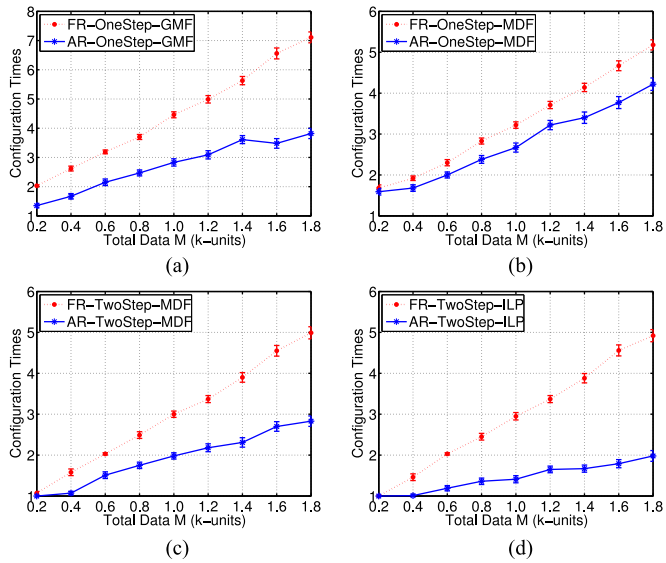

Fig. 6. Comparison on reconfiguration times from *AR-based* and *FR-based* heuristics when $\Delta t = 1$ (time-unit).

1) Results on DC-B-Wnd and Reconfiguration Times: Fig. 5 shows the results on DC-B-Wnd from the FR- and AR-based algorithms. We observe that when we use *OneStep-GMF*, *OneStep-MDF* or *TwoStep-MDF* as the sub-routine to determine the Backup Scenario for each Δt, the corresponding AR-based algorithms provide longer DC-B-Wnds than the FR-based counterparts. This is because AR-based algorithms adjust the Backup Scenario adaptively but not for each Δt. Note that the results on DC-B-Wnd from *AR-OneStep-GMF* are abnormally long with relatively large confidence intervals. This phenomenon can be explained as follows. *OneStep-GMF* optimizes the Backup Scenario mainly based on the network bandwidth resources, but does not pay much attention on the remaining data to be backed up on each DC. Hence, the scheme may lead to the situation in which less bandwidth is allocated to a DC with more data to be backed up. This can bring up the randomness on results, which causes the large confidence intervals. Moreover, as Lines 3–8 in *Algorithm* 2 may decide to invoke the reconfigurations on the Backup Scenario with much less frequency, we end up with severely prolonged DC-B-Wnds. Meanwhile, we notice that if we use *TwoStep-ILP* as the sub-routine, *AR-TwoStep-ILP* and *FR-TwoStep-ILP* show similar results on DC-B-Wnd.

Fig. 6 plots the results on the reconfiguration times from the FR- and AR-based algorithms. As expected, the AR-based algorithms invoke less reconfigurations on the Backup Scenario than the FR-based ones. We also observe that if we keep the same sub-routine but change the algorithm from FR-based to AR-based, the reconfiguration times decrease more for the cases using *OneStep-GMF* and *TwoStep-ILP* than those that incorporate *OneStep-MDF* and *TwoStep-MDF*. Moreover, by combining the results in Figs. 5 and 6, we can see that *AR-TwoStep-ILP* requires much less reconfiguration times than *FR-TwoStep-ILP*, but their performance on DC-B-Wnd is similar. This is because *TwoStep-ILP* can handle the data-transfers of all the backup pairs simultaneously and the data on the DCs is prone to decrease proportionally. This leads to relatively infrequent changes on the data ranks, and hence makes some of the reconfigurations invoked by *FR-TwoStep-ILP* become unnecessary when using *AR-TwoStep-ILP*.

TABLE IV
AVERAGE COMPUTATION TIME IN SECONDS FOR $\Delta t = 1$ TIME-UNIT

Algorithms	Total Backup Data M (k-units)				
	1.2	1.4	1.6	1.8	2
FR-OneStep-GMF	1.6355	1.8322	2.0437	2.2007	2.422
FR-OneStep-MDF	0.4627	0.538	0.586	0.6666	0.7015
FR-TwoStep-MDF	0.13	0.1476	0.168	0.1868	0.207
FR-TwoStep-ILP	0.1366	0.1577	0.1738	0.1958	0.2136
AR-OneStep-GMF	1.3903	1.5322	1.7151	1.8039	1.9401
AR-OneStep-MDF	0.4132	0.4568	0.5055	0.5536	0.5962
AR-TwoStep-MDF	0.1093	0.1172	0.1302	0.1458	0.1535
AR-TwoStep-ILP	0.0892	0.0983	0.1031	0.1058	0.1268

To this end, we can conclude that *AR-TwoStep-ILP* achieves the best tradeoff between DC-B-Wnd and the operational complexity, among all the heuristics.

2) Results on Computation Time: Table IV presents the results on the average computation time of different heuristics. The results indicate that the AR-based algorithms consume shorter computation time than their FR-based counterparts, since the AR-based ones adaptively determine when to reconfigure the Backup Scenario. Moreover, we observe that *AR-TwoStep-ILP* takes shorter computation time than *AR-TwoStep-MDF*, which makes it the fastest heuristics among the eight. This is because when changing the algorithm from FR-based to AR-based, *TwoStep-ILP* achieves more reductions on the reconfiguration times than *TwoStep-MDF*, and hence *AR-TwoStep-ILP* becomes more time-efficient.

In all, the simulation results indicate that *AR-TwoStep-ILP* is the most time-efficient heuristic to design the fast and coordinated data backup in the optical inter-DC networks, and it

achieves the best tradeoff between DC-B-Wnd and operational complexity.

VII. Conclusion

In this paper, we investigated fast and coordinated data backup in optical inter-DC networks. By considering a mutual backup model, we studied how to minimize the DC backup window (DC-B-Wnd) with joint optimization of backup site selection and data-transfer paths. An ILP was first formulated and then a few heuristics were proposed to reduce the computation time. Moreover, in order to explore the tradeoff between DC-B-Wnd and operational complexity, we proposed heuristics based on AR. Simulation results indicated that among all the proposed heuristics, *AR-TwoStep-ILP* achieves the best tradeoff between DC-B-Wnd and operational complexity and it is also the most time-efficient one. Therefore, the best policy to follow in this problem is to obtain backup site selection and data-transfer paths in two separate steps.

References

[1] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for Internet-scale systems," in *Proc. ACM SIGCOMM*, Aug. 2009, pp. 1–12.

[2] A. Greenberg, J. Hamilton, D. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, pp. 68–73, Jan. 2009.

[3] A. Krishnan and R. Sitaraman, "Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs," in *Proc. Internet Meas. Conf.*, Nov. 2012, pp. 1–14.

[4] C. Liu, A. Kind, and A. Vasilakos, "Sketching the data center nework traffic," *IEEE Netw.*, vol. 27, no. 4, pp. 33–39, Jul./Aug. 2013.

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

[6] K. Georgakilas, A. Tzanakaki, M. Anastasopoulos, and J. Pedersen, "Converged optical network and data center virtual infrastructure planning," *J. Opt. Commun. Netw.*, vol. 4, pp. 681–691, Sep. 2012.

[7] [Online]. Available: http://en.wikipedia.org/wiki/2008_Sichuan_earthquake

[8] [Online]. Available: http://en.wikipedia.org/wiki/2011_Tohoku_earthquake_and_tsunami

[9] S. Frolund and F. Pedone, "Dealing efficiently with data-center disasters," *J. Parallel Distrib. Comput.*, vol. 63, pp. 1064–1081, Nov. 2003.

[10] J. Mehr, E. Murphy, N. Virk, and L. Sosnosky, "Hybrid distributed and cloud backup architecture," U.S. Patent 20 100 274 982, Apr. 2009.

[11] Y. Song, R. Routray, and Y. Hou, "Scalable data analytics platform for enterprise backup management," in *Proc. Netw. Oper. Manag. Symp.*, May 2014, pp. 1–7.

[12] N. Mandagere, P. Zhou, M. Smith, and S. Uttamchandani, "Demystifying data deduplication," in *Proc. USENIX Middleware*, Dec. 2008, pp. 7–12.

[13] H. Chan and T. Chieu, "An approach to high availability for cloud servers with snapshot mechanism," in *Proc. USENIX Middleware*, Dec. 2012, pp. 1–6.

[14] W. Xia, H. Jiang, D. Feng, and L. Tian, "Combining deduplication and delta compression to achieve low-overhead data reduction on backup datasets," in *Proc. Data Compression Conf.*, Mar. 2014, pp. 203–212.

[15] A. Garcia-Recuero, S. Esteves, and L. Veiga, "Quality-of-data for consistency levels in geo-replicated cloud data stores," in *Proc. CloudCom*, Dec. 2013, pp. 164–170.

[16] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Zomaya, "Energy-efficient data replication in cloud computing datacenters," *Cluster Comput.*, vol. 18, pp. 385–402, Jan. 2015.

[17] [Online]. Available: https://en.wikipedia.org/wiki/Replication_(computing)

[18] J. Zheng and H. Mouftah, "Routing and wavelength assignment for advance reservation in wavelength-routed WDM optical networks," in *Proc. Int. Conf. Commun.*, Aug. 2002, pp. 2722–2726.

[19] D. Andrei, H. Yen, M. Tornatore, C. Martel, and B. Mukherjee, "Integrated provisioning of sliding scheduled services over WDM optical networks," *J. Opt. Commun. Netw.*, vol. 1, pp. A94–A105, Jul. 2009.

[20] A. Patel and J. Jue, "Routing and scheduling for variable bandwidth advance reservation," *J. Opt. Commun. Netw.*, vol. 3, pp. 912–923, Dec. 2011.

[21] N. Charbonneau and V. Vokkarane, "A survey of advance reservation routing and wavelength assignment in wavelength-routed WDM networks," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 1037–1064, Oct.-Dec. 2012.

[22] J. Yao, P. Lu, and Z. Zhu, "Minimizing disaster backup window for geo-distributed multi-datacenter cloud systems," in *Proc. Int. Conf. Commun.*, Jun. 2014, pp. 1–5.

[23] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-datacenter bulk transfers with netstitcher," in *Proc. ACM SIGCOMM*, Aug. 2011, pp. 74–85.

[24] A. Mahimkar, A. Chiu, R. Doverspike, M. Feuer, P. Magill, E. Mavrogiorgis, J. Pastor, S. Woodward, and J. Yates, "Bandwidth on demand for inter-data center communication," in *Proc. ACM HotNets*, Nov. 2011, pp. 24–29.

[25] Oracle White Paper: Backup is not archiving: Reduce TCO and improve data protection with a customized archive solution. [Online]. Available: http://www.oracle.com/us/products/servers-storage/storage/tape-storage/backupnotarchivefinal-120513gc-2083314.pdf

[26] C. Develder, J. Buysse, B. Dhoedt, and B. Jaumard, "Joint dimensioning of server and network infrastructure for resilient optical grids/clouds," *IEEE/ACM Trans. Netw.*, vol. 22, no. 5, pp. 1–16, Oct. 2014.

[27] M. Habib, M. Tornatore, M. Leenheer, F. Dikbiyik, and B. Mukherjee, "Design of disaster-resilient optical datacenter networks," *J. Lightw. Technol.*, vol. 30, no. 16, pp. 2563–2573, Aug. 2012.

[28] J. Zhang, Y. Zhao, H. Yang, Y. Ji, H. Li, Y. Lin, G. Li, J. Han, Y. Lee, and T. Ma, "First demonstration of enhanced software defined networking (eSDN) over elastic grid (eGrid) optical networks for data center service migration," in *Proc. Opt. Fiber. Conf.*, Mar. 2013, pp. 1–3.

[29] Y. Katsuyama, M. Hashimoto, K. Nishikawa, A. Ueno, M. Nooruzzaman, and O. Koyama, "Lightpath reconfiguration in regional IP-over-WDM networks by a centralized control system," in *Proc. Local Comput. Netw.*, Oct. 2007, pp. 63–72.

[30] S. Naiksatam, S. Figueira, S. Chiappari, and N. Bhatnagar, "Analyzing the advance reservation of lightpaths in lambda-grids," in *Proc. IEEE Cluster Comput. Grid*, May 2005, pp. 985–992.

[31] L. Shen, X. Yang, A. Todimala, and B. Ramamurthy, "A two-phase approach for dynamic lightpath scheduling in WDM optical networks," in *Proc. Int. Conf. Commun.*, Jun. 2007, pp. 2412–2417.

[32] Y. Chen, A. Jaekel, and A. Bari, "Resource allocation strategies for a non-continuous sliding window traffic model in WDM networks," in *Proc. BROADNETS*, Sep. 2009, pp. 1–7.

[33] V. Winkler,*Securing the Cloud*. New York, NY, USA: Elsevier, 2011.

[34] W. Jia, D. Xuan, and W. Zhao, "Integrated routing algorithms for anycast messages," *IEEE Commun. Mag.*, vol. 38, no. 1, pp. 48–53, Jan. 2000.

[35] M. Gharbaoui, B. Martini, and P. Castoldi, "Anycast-based optimizations for inter-data-center interconnections," *J. Opt. Commun. Netw.*, vol. 4, pp. B168–B178, Nov. 2012.

[36] T. Cormen, C. Leiserson, R. Rivest, and C. Stein,*Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2009.

[37] D. West, *Introduction to Graph Theory*. Englewood Cliffs, NJ, USA: Prentice-Hall, Aug. 2000.

[38] A. Schrijver *Theory of Linear and Integer Programming*. New York, NY, USA: Wiley, 1998.

[39] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *Proc. ACM Symp. Theory Comput.*, May 1984, pp. 302–311.

[40] [Online]. Available: http://www.slideshare.net/kmstechnology/big-data-overview-2013-2014

[41] J. Dean and S. Ghmawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, pp. 107–113, Jan. 2008.

Authors' biographies not available at the time of publication.