Maximizing Utility of Time-Constrained Emergency Backup in Inter-Datacenter Networks

Ping Lu, Qing Ling, and Zuqing Zhu, *Senior Member, IEEE*

***Abstract*—This work addresses the emergency backup problem in inter-datacenter (inter-DC) network, which is triggered in response to predictable destructive events. We consider the value of data and study how to maximize the utility of time-constrained emergency backup. Specifically, we first formulate the utility-maximization problem for a time-varying network environment and then leverage the time-expanded network (TEN) approach to transform it into a static network problem. To finally solve the problem, a time-efficient and distributed algorithm is proposed based on the dual decomposition technique. Simulation results show that our algorithm outperforms existing algorithms.**

***Index Terms*—Time-constrained data backup, Inter-datacenter networks, Time-expanded networks, Dual decomposition.**

I. Introduction

NOWADAYS, with the fast development of data-intensive applications, we have entered the era of Big Data and data is treated as valuable asset by various enterprises now. Hence, numerous datacenters (DCs) have been built recently and DC networking has become an attracting research topic [1]. Meanwhile, as DCs can carry massive data and serve millions of applications, they have to incorporate data backup to obtain sufficient data redundancy and prevent the service interruptions due to natural disasters and human misconduct. Specifically, enterprises may own or rent several geographically distributed DCs and invoke mutual data backup among them [2]. In such inter-DC networks, data backup bears two forms, *i.e.*, regular backup and emergency backup.

Previously, researches have studied regular backup and proposed schemes to either reduce the resource costs [3] or shorten the backup time window [2]. However, emergency backup that is triggered in response to predictable destructive events (*e.g.*, flood, tsunami and missile attack) is also very relevant but has not been fully explored yet. Recently, in [4], Ma *et al.* treated all the endangered data equally and investigated how to minimize the overall resource costs for time-constrained emergency backup in inter-DC networks. Note that, in the era of Big Data, data itself is multi-modal and has differentiated values [1], and thus it should not be simply treated as equivalent bits in data transfers. This is especially true for emergency backup, since we might not be able to duplicate all the endangered data out within a limited time window. Hence, for emergency backup, we need to prioritize the endangered data sets according to their values and try to maximize the data owners' utilities.

Manuscript received February 28, 2016; revised March 21, 2016; accepted March 24, 2016. Date of publication March 28, 2016; date of current version May 6, 2016. This work was supported in part by the NSFC under Grants 61371117 and 61573331, Grant WK2100060010, and Grant KJ2014ZD38 and in part by the SPR Program of CAS (XDA06011202). The associate editor coordinating the review of this paper and approving it for publication was T. Han.

The authors are with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China (e-mail: zqzhu@ieee.org).

Digital Object Identifier 10.1109/LCOMM.2016.2547427

In this work, we study how to maximize the utility of time-constrained emergency backup in inter-DC networks. We first consider the completeness of data and model the value of backed-up data based on its size. Then, we formulate the utility-maximization problem for a time-varying network environment and leverage the time-expanded network (TEN) approach [5] to transform it into a static network problem. Finally, a time-efficient and distributed backup algorithm is proposed with the assistance of dual decomposition. Simulation results show that our proposed algorithm can converge to the optimal solution and outperform several existing algorithms.

The rest of the letter is organized as follows. In Section II, we formulate the problem of maximizing the utility of time-constrained emergency backup. Section III transforms the dynamic network problem into a static one with the TEN approach, and the distributed algorithm is proposed in Section IV. We evaluate the proposed algorithm in Section V. Finally, Section VI summarizes the letter.

II. Problem Formulation

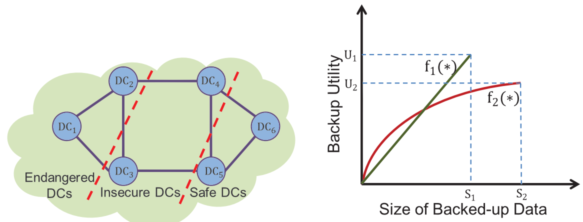
A. Network Model

We denote an inter-DC network as $\mathcal{G} = (\mathcal{D}, \mathcal{E})$, where $\mathcal{D} = \{1, \cdots, |\mathcal{D}|\}$ is the DC set and $\mathcal{E}$ is the set of links that connect the DCs. To model the time-varying network environment, we consider a discrete-time system in which all the operations are performed at $t = \Delta t, 2\Delta t, \cdots$. These time intervals can be represented as $t = 1, 2, \cdots$, if we normalized them with Δt. The available bandwidth on link $e \in \mathcal{E}$ at time t is $B_e(t)$.

When a predictable destructive event is about to happen on one or more DCs, we estimate the time window for backing up the endangered data and try to fully utilize it to maximize the data owners' utilities. The duration of the time window is assumed as T, which means that emergency backup can operate contiguously at time intervals $\mathcal{T} = \{1, \cdots, T\}$. As the upcoming destructive event's impact on the DCs can also be predicted or estimated, we classify the DCs into the following three categories accordingly, as shown in Fig. 1(a).

- *Endangered DCs*: They are denoted as $\mathcal{D}_{en}$, and will be destroyed by the upcoming event. For a DC $i \in \mathcal{D}_{en}$, the amount of data that needs to be backed up is S_i.
- *Insecure DCs*: They are denoted as $\mathcal{D}_{in}$, and may be impacted by the upcoming event. Hence, these DCs cannot be used as the destinations of emergency backup, but they can work as the intermediate nodes of data transfers and buffer the endangered data[1]. We denote the available storage space on a DC $i \in \mathcal{D}_{in}$ as C_i.
- *Safe DCs*: They are denoted as $\mathcal{D}_d$, and will not be impacted by the upcoming event. The emergency backup

[1]Note that, since the time window for emergency backup is very limited, we assume that the original data in insecure DCs can be backed up later.



(a) Classification of DCs. (b) Utility of backed up data.

Fig. 1. Network model of time-constrained emergency backup.

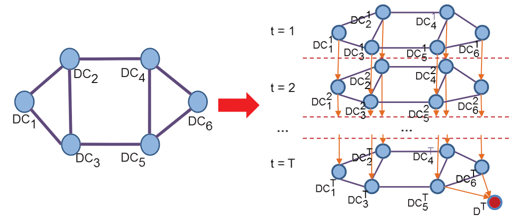


Fig. 2. Example of time-expanded network (TEN) approach.

can use any DC $i \in \mathcal{D}_d$ as a destination and the available storage space on the DC is also C_i.

B. Utility Function of Emergency Backup

Apparently, the utility of emergency backup should depend on the size of the backed-up data. The utility function of an endangered DC $i \in \mathcal{D}_{en}$ is defined as $f_i(s)$, where s is the total backed-up data that is successfully received on all the safe DCs in $\mathcal{D}_d$ within time window T. Here, we assume that each endangered DC backs up its data in descending order of the value. Therefore, the utility function $f_i(\cdot)$ will only have one of the two forms that are shown in Fig. 1(b). If all the endangered data on DC i has the same value, the utility increases linearly with the size of the backed-up data (*i.e.*, indicated by $f_1(\cdot)$). Otherwise, the utility function has the form of $f_2(\cdot)$, since the data that is backed up later has lower value.

C. Utility-Maximizing Problem

To model emergency backup, we assign a few queues on each DC in the inter-DC networks and use variable $Q_m^i(t)$ to denote the corresponding queue length, *i.e.,* the amount of data that was originally from DC $i \in \mathcal{D}_{en}$ but is currently buffered in DC $m \in \mathcal{D}$ at time t. Hence, initially, we have

$$\begin{cases} Q_m^i(t=1) = 0, \ \forall m \in \mathcal{D} \setminus \mathcal{D}_{en}, \\ Q_m^i(t=1) = S_i, \ \forall m \in \mathcal{D}_{en}. \end{cases} \tag{1}$$

Then, in the whole emergency backup process, the total queue length in each insecure or safe DC should not exceed the DC's available storage space, and we have

$$\sum_{i \in \mathcal{D}_{en}} Q_m^i(t) \le C_m, \ \forall m \in \mathcal{D} \setminus \mathcal{D}_{en}, \ \forall t \in \mathcal{T}. \tag{2}$$

We also introduce a variable $x_{m,n}^i(t)$ to represent the amount of data that is transferred from $Q_m^i(t)$ to $Q_n^i(t)$ at time t. Apparently, the transferred data should not exceed the buffered data in DC m, *i.e.*,

$$\sum_{n \in \mathcal{D}} x_{m,n}^i(t) \le Q_m^i(t), \ \forall m \in \mathcal{D}, i \in \mathcal{D}_{en}, \ \forall t. \tag{3}$$

Then, the queue update rule becomes

$$\begin{aligned} Q_m^i(t+1) = \min &\left\{ Q_m^i(t) - \sum_n x_{m,n}^i(t), 0 \right\} \\ &+ \sum_n x_{n,m}^i(t), \quad \forall i, m \in \mathcal{D}, \ \forall t. \end{aligned} \tag{4}$$

We use $b_e^{m,n,i}(t)$ to represent the bandwidth allocation on link e for data transfer $x_{m,n}^i(t)$, and the flow conservation constraint for each DC u is

$$\sum_{e \in \mathcal{O}(u)} b_e^{m,n,i}(t) - \sum_{e \in \mathcal{I}(u)} b_e^{m,n,i}(t) = \begin{cases} x_{m,n}^i(t), & u = m, \\ 0, & \text{otherwise}, \ \forall u \in \mathcal{D}, \ \forall t, \\ -x_{m,n}^i(t), & u = n, \end{cases} \tag{5}$$

where $\mathcal{O}(u)$ and $\mathcal{I}(u)$ are the sets of the links that origin from and end at DC u, respectively. And all the data transfers should follow the bandwidth constraint, *i.e.*,

$$\sum_{i \in \mathcal{D}_{en}} \sum_{m,n \in \mathcal{D}} b_e^{m,n,i}(t) \le B_e(t), \ \forall e \in \mathcal{E}. \tag{6}$$

Finally, with the network model mentioned above, we can formulate the problem of maximizing the utility of time-constrained emergency backup as

$$\begin{aligned} \text{Maximize} \quad & \sum_{i \in \mathcal{D}_{en}} f_i \left(\sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{D}_d} x_{m,n}^i(t) \right) \\ s.t. \quad & \text{Eqs. (1)} - (6). \end{aligned} \tag{7}$$

However, we hope to point out that the optimization problem in Eq. (7) is complex and can even become intractable due to the fact that the numbers of variables and constraints increase quickly with the scale of the problem, *i.e.*, $|\mathcal{D}|$, $|\mathcal{E}|$, and $|\mathcal{T}|$.

III. Problem Transformation With TEN Approach

In order to reduce the time complexity of the optimization in Eq. (7), we leverage the time-expanded network (TEN) approach discussed in [5]. Specifically, we first replicate the network along the time axis for $T-1$ times. As shown in Fig. 2, replica t represents the network status at time t and is denoted as $\mathcal{G}^t = (\mathcal{D}^t, \mathcal{E}^t)$. The bandwidth of a link inside replica t is just the link's available bandwidth at time t. To represent the available DC storage that can be used to buffer data, we add a directed link from each DC i in replica t to the same DC in replica $t+1$. Hence, the bandwidth of a link in between replicas is the corresponding DC's available storage space at time t. Next, we insert a super destination D^T in replica T, which receives a directed link from each safe DC in the replica. The virtual bandwidth on the link equals the storage space of the DC from which it origins. Finally, we transform the original optimization problem into the one to find the utility-maximized multi-commodity flow (UM-MCF) in the TEN, *i.e.*, how to schedule and route the data transfers from the DCs in $\mathcal{D}_{en}$ to the super destination D^T to obtain the maximal utility. Hence, the TEN approach transforms the dynamic network problem into a static one.

We denote the topology of the TEN as $\mathcal{G}^* = (\mathcal{D}^*, \mathcal{E}^*)$, where $\mathcal{D}^*$ and $\mathcal{E}^*$ are the node and link sets, respectively.

The bandwidth on a link $e \in \mathcal{E}^*$ is B_e^*. We introduce variable x_i to represent the total amount of the backup flow from an endangered DC i to the super destination D^T, while b_e^i indicates the bandwidth allocation on link $e \in \mathcal{E}^*$ for the backup flow from the DC. The amount of the backup flow should not exceed the total amount of data for backup, *i.e.*,

$$x_i \leq S_i, \quad \forall i \in \mathcal{D}_{en}, \tag{8}$$

and the flows on each link satisfy the bandwidth constraint

$$\sum_{i \in \mathcal{D}_{en}} b_e^i \leq B_e^*, \quad \forall e \in \mathcal{E}^*. \tag{9}$$

The flow conservation constraint uses parameter x_u^i as

$$x_u^i = \begin{cases} 1, & u = i, \\ 0, & \text{otherwise}, \ \forall i \in \mathcal{D}_{en},\ u \in \mathcal{D}^*, \\ -1, & u = D^T. \end{cases} \tag{10}$$

and introduces a parameter $M_{u,e}$ to indicate the relation between a DC u and a link e. Specifically, we have $M_{u,e} = 1$ if $e \in \mathcal{O}_u$, or we have $M_{u,e} = -1$ if $e \in \mathcal{I}_u$, or $M_{u,e} = 0$, otherwise. Then, the flow conservation constraint is

$$\sum_{e \in \mathcal{E}^*} M_{u,e} \cdot b_e^i = x_u^i \cdot x_i,\ \forall i \in \mathcal{D}_{en},\ u \in \mathcal{D}^*. \tag{11}$$

Finally, we can formulate the UM-MCF problem for maximizing the utility of time-constrained emergency backup as

$$\begin{aligned} \text{Maximize} \quad & \sum_{i \in \mathcal{D}_{en}} f_i(x_i) \\ s.t. \quad & \text{Eqs. (8)} - \text{(11)}. \end{aligned} \tag{12}$$

IV. Distributed Algorithm Based on Dual Decomposition

We find that variables x_i and b_e^i are coupled in Eq. (11). By leveraging Lagrange relaxation, we can decompose the UM-MCF problem in Eq. (12) and solve it time-efficiently. With these considerations, we design a distributed algorithm.

We introduce dual variables $\{\lambda_{i,u}, \forall i \in \mathcal{D}_{en}, u \in \mathcal{D}^*\}$ and write the Lagrangian function of Eq. (12) as

$$\begin{aligned} L(\cdot) &= \sum_{i \in \mathcal{D}_{en}} \left[f_i(x_i) + \sum_{u \in \mathcal{D}^*} \lambda_{i,u} \cdot \left(\sum_{e \in \mathcal{E}^*} M_{u,e} \cdot b_e^i - x_u^i \cdot x_i \right) \right] \\ &= \sum_{i \in \mathcal{D}_{en}} \left(f_i(x_i) - \sum_{u \in \mathcal{D}^*} \lambda_{i,u} \cdot x_u^i \cdot x_i \right) \\ &\quad + \sum_{i \in \mathcal{D}_{en}} \sum_{u \in \mathcal{D}^*} \sum_{e \in \mathcal{E}^*} \lambda_{i,u} \cdot M_{u,e} \cdot b_e^i, \end{aligned} \tag{13}$$

where $L(\cdot) = L(\{x_i\}, \{b_e^i\}, \{\lambda_{i,u}\})$ and the primal variables are subjected to the constraints Eqs. (8)–(10). If we define the Lagrange function as $G(\{\lambda_{i,u}\})$, the dual problem becomes

$$\begin{aligned} \text{Minimize} \quad & G(\{\lambda_{i,u}\}) = \max_{\{x_i\},\{b_e^i\}} [L(\cdot)] \\ s.t. \quad & \text{Eqs. (8)} - \text{(10)}. \end{aligned} \tag{14}$$

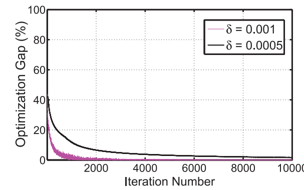


Fig. 3. Overall optimization gap with $T = 5$ minutes.

The dual problem in Eq. (14) can be decomposed into two independent subproblems. Firstly, for each endangered DC $i \in \mathcal{D}_{en}$, we have the subproblem for getting backed-up data

$$\begin{aligned} \text{Maximize} \quad & f_i(x_i) - \sum_{u \in \mathcal{D}^*} \lambda_{i,u} \cdot x_u^i \cdot x_i \\ s.t. \quad & x_i \leq S_i. \end{aligned} \tag{15}$$

Here, if we adopt a logarithmic utility function [6] as $f_i(x_i) = \alpha_i \cdot \log(1 + x_i)$, where α_i is the utility weight, and notice the definition of x_u^i, Eq. (15) can be rewritten as

$$\begin{aligned} \text{Maximize} \quad & \alpha_i \cdot \log(1 + x_i) - (\lambda_{i,i} - \lambda_{i,D^T}) \cdot x_i \\ s.t. \quad & x_i \leq S_i. \end{aligned} \tag{16}$$

With $\widehat{\lambda_i} = (\lambda_{i,i} - \lambda_{i,D^T}) \cdot \ln(10)$, the optimal value of x_i is

$$x_i^* = \begin{cases} 0, & \widehat{\lambda_i} > \alpha_i, \\ \frac{\alpha_i}{\widehat{\lambda_i}} - 1, & \frac{\alpha_i}{1+S_i} < \widehat{\lambda_i} \leq \alpha_i, \\ S_i, & \widehat{\lambda_i} \leq \frac{\alpha_i}{1+S_i}. \end{cases} \tag{17}$$

Basically, dual variables $\{\lambda_{i,u}\}$ can be viewed as the potentials on DCs for emergency backup, which are determined based on the data for backup and utility functions. Hence, the solution in Eq. (17) indicates that each endangered DC calculates the total amount of its backup flow according to the difference between its potential and that on the super destination D^T.

Secondly, the subproblem for getting bandwidth allocation on each link $e \in \mathcal{E}^*$ for backup flows is

$$\begin{aligned} \text{Maximize} \quad & \sum_{i \in \mathcal{D}_{en}} (\lambda_{i,e^-} - \lambda_{i,e^+}) \cdot b_e^i \\ s.t. \quad & \sum_{i \in \mathcal{D}_{en}} b_e^i \leq B_e^*, \end{aligned} \tag{18}$$

where e^- and e^+ are the end nodes of e, *i.e.*, $e = (e^-, e^+)$. We can see that the optimization in Eq. (18) is a standard knapsack problem. Hence, all the bandwidth on e should be allocated to the backup flow whose potential difference between e^- and e^+ is the maximum, *i.e.*, the flow for DC $\hat{i} = \text{argmax}_{i \in \mathcal{D}_{en}}(\lambda_{i,e^-} - \lambda_{i,e^+})$. Then, the optimal bandwidth allocation is

$$(b_e^i)^* = \begin{cases} B_e^*, & i = \hat{i}, \\ 0, & \text{otherwise}. \end{cases} \tag{19}$$

The dual master problem in Eq. (14) can be solved using the sub-gradient method. Specifically, we compute $\{\lambda_{i,u}\}$ in iterations such that $G(\{\lambda_{i,u}\})$ converges to the minimum. In the $(k+1)$-th iteration, we get dual variable $\lambda_{i,u}^{k+1}$ as follows.

$$\lambda_{i,u}^{k+1} = \lambda_{i,u}^k - \delta_k \cdot \left(\sum_{e \in \mathcal{E}^*} M_{u,e} \cdot (b_e^i)^k - (x_u^i)^k \right), \tag{20}$$

where δ_k is the step-size in the k-th iteration. We adopt $\delta_k = \delta/\sqrt{k}$, where δ is an adjustable coefficient.

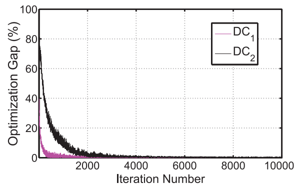


Fig. 4. Optimization gaps of endangered DCs with $T = 5$ minutes and $\delta = 0.001$.

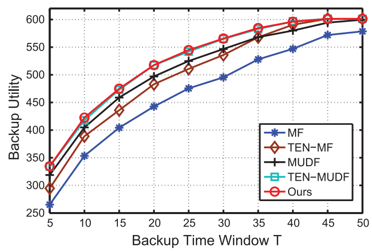


Fig. 5. Comparisons on backup utilities.

TABLE I
RUNNING TIME (SECONDS)

T	MF	TEN-MF	MUDF	TEN-MUDF	Ours
15	0.029	0.040	0.034	1.080	0.112
30	0.054	0.105	0.068	7.749	0.147
45	0.094	0.207	0.126	21.443	0.220

V. PERFORMANCE EVALUATION

We perform numerical simulations to evaluate the proposed distributed algorithm. Here, we consider a real inter-DC network topology in USA [7], which consists of 11 DCs and 34 directed links. The available bandwidth on each link is randomly chosen within [40, 400] Gbps. The simulations assume that a destructive events will destroy two DCs (*i.e.*, two endangered DCs) after a time-window T and the other two DCs would become insecure ones due to the event, while the rest DCs are safe. The DCs are categorized accordingly based on their geographical locations in the topology. For each endangered DC, we have 100 TBytes data that need to be backed up. The logarithmic utility functions of the endangered DCs use different weights as $\alpha_1 = 100$ units and $\alpha_2 = 200$ units, respectively. The storage space on each insecure or safe DC is set randomly, but their total storage space is fixed as 500 TBytes. We choose the time interval as $\Delta t = 1$ minute, as it is enough for the network operation change on each DC and the information exchange among the DCs.

We first verify that the result from our distributed algorithm can converge to that of the primal problem, which is obtained by solving the optimization in Eq. (12) in a centralized manner directly. Fig. 3 plots the optimization gap on the backup utility from the distributed algorithm, when we have $T = 5$ minutes. Here, we test two values of the adjustable coefficient δ for the step-size, *i.e.*, $\delta = 0.001$ and 0.0005. It can be seen that the gap decreases with the iterations and eventually converges to 0 for both cases. Meanwhile, we notice that the convergence speed can be affected by the step-size and Fig. 3 shows that the convergence is much faster when we have $\delta = 0.001$, *i.e.*, using a larger step-size. Here, we hope to point out that the step-size cannot be arbitrarily large, as an overlarge step-size can prevent the algorithm from converging. Furthermore, we study the optimization gaps on the backup utilities of the two endangered DCs with $\delta = 0.001$, and the results are plotted in Fig. 4. We observe that the gaps also converge to 0.

Next, we compare our algorithm with several existing ones. First of all, inspired by [2], we consider a maximum flow (MF) based algorithm, which performs data transfers by using the multi-source multi-destination maximum flow in the network in each time interval. Secondly, we consider the straight-forward idea to back up data with higher utility earlier and develop a maximum-utility-data-first (MUDF) algorithm. Basically, in each time interval, we sort the endangered DCs based on the utilities of the left-over data on them in descending order and then perform data transfers for them one by one with the maximum flows. Thirdly, we introduce the TEN approach to the MF and MUDF algorithms and design two more benchmarks as TEN-MF and TEN-MUDF, respectively.

Fig. 5 compares the backup utilities obtained by the algorithms. We observe that the utility increases with T for all the algorithms. In general, since the TEN-based algorithms can utilize the storage space on insecure DCs more efficiently, they achieve higher backup utilities than their counterparts without the TEN approach. Our algorithm achieves the highest utility among all the algorithms, while the utility from TEN-MUDF is comparable. The running time of the algorithms is listed in Table I[2]. It can be seen that our algorithm runs much faster than TEN-MUDF and its running time is similar to that of the other three benchmarks.

VI. CONCLUSION

This work studied how to maximize the utility of time-constrained emergency backup in inter-DC networks. We proposed a time-efficient and distributed backup algorithm. Simulation results showed that our proposed algorithm outperformed several existing algorithms.

REFERENCES

[1] P. Lu *et al.*, "Highly-efficient data migration and backup for big data applications in elastic optical inter-datacenter networks," *IEEE Netw.*, vol. 29, no. 5, pp. 36–42, Sep./Oct. 2015.

[2] Y. Yao, P. Lu, L. Gong, and Z. Zhu, "On fast and coordinated data backup in geo-distributed optical inter-datacenter networks," *J. Lightw. Technol.*, vol. 33, no. 4, pp. 3005–3015, Jul. 2015.

[3] T. Wood *et al.*, "Disaster recovery as a cloud service: Economic benefits & deployment challenges," in *Proc. USENIX HotCloud*, Jun. 2010, pp. 1–7.

[4] L. Ma, X. Jiang, B. Wu, and T. Taleb, "Cost-efficient data backup for data center networks against ϵ-time early warning disaster," Dec. 2015 [Online]. Available: http://arxiv.org/pdf/1512.08189v2.pdf

[5] L. Ford and D. Fulkerson, "Constructing maximal dynamic flows from static flows," *Oper. Res.*, vol. 6, pp. 419–433, Jun. 1958.

[6] M. Chiang, S. Low, A. Calderbank, and J. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," *Proc. IEEE*, vol. 95, no. 1, pp. 255–312, Jan. 2007.

[7] Y. Wang *et al.*, "Multiple bulk data transfers scheduling among datacenters," *Comput. Netw.*, vol. 68, pp. 123–137, Aug. 2014.

[2]All the simulations run on a computer with an Intel CPU (I3-2120, 3.30GHz) and 8 GB memory.