D4D: Inter-Datacenter Bulk Transfers with ISP Friendliness

Yangyang Li, Hongbo Wang, Peng Zhang, Jiankang Dong, Shiduan Cheng
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China, 100876
Email: {yyli, hbwang, zhp, dongjk, chsd}@bupt.edu.cn

***Abstract*—Cloud infrastructure providers (CIPs) usually build a large number of geo-distributed datacenters to cater to the recent proliferation of Cloud Computing. The CIPs commonly use multiple internet service providers (ISPs) to interconnect their geo-distributed datacenters. Diurnal patterns and leftover bandwidth were effectively exploited by past studies to improve bandwidth utilization and minimize cost on inter-datacenter traffic. However, the growing number of inter-domain traffic was neglected. Inter-domain communications may become the potential bottleneck of inter-datacenter bulk transfers. Moreover, the rising inter-domain traffic increases the ISP operational cost, which will be not beneficial to reduce the CIPs' bandwidth cost or improve the quality of services over a long run. In this paper, we present a scheduling scheme that considers both bandwidth utilization and ISPs friendliness via a store-and-forward mechanism. The problem is modeled on a time-expanded graph. We compared our scheme with general bulk transfers mechanism under several different simulation settings. The results demonstrate that our strategy reduces the inter-domain traffic tremendously and achieves the ISP-friendliness significantly.**

***Keywords*—bulk transfers, inter-datacenter traffic, ISP friendly**

I. Introduction

In recent years, the fast proliferation of Cloud Computing has incited Cloud infrastructure providers (CIPs) to build a large number of datacenters distributed around the world. Not only cloud providers such as Google, Mircosoft and Amazon replicate user data across geographical regions, but also content delivery providers, for instance, Akamai and ChinaCache widely invest in distributed internet datacenters to respone client's request as soon as possible. Geo-diversifying datacenters can improve end to end performance and increase reliability in the event of site failures [1]. However, applications continue to become more data-intense, when application is decomposed across geographical datacenters, this may complicates data placement and transport [2]. For this reason, inter-datacenter traffic has got much attention recently.

Inter-datacenter traffic contributes to a large portion of datacenters' export traffic. In [3], Chen *et al*. investigated inter-datacenter traffic characteristics via five Yahoo! datacenters. They found that D2D (Datacenter-Datacenter, differ from transit Datacenter-Client) traffic, including backups, disaster recovery replication, regional Vmotion and recent trends in big data analysis, occupies near half of the inter-datacenter traffic. In order to cope with the peak traffic demand, CIPs always purchase over-subscribed bandwidth for applications' requirement. But datacenters distribute across time zones and services spread unevenly across geo-distributed datacenters, causing the links between datacenters are used ineffectively. Since most of the D2D jobs are time in-sensitive, recently, many research proposed to use intermediate datacenters temporarily store data and forward it in a later time. It is an effective mechanism to improve the utilization of inter-datacenter overlay links [4] and reduce the cost on inter-datacenter traffic [5] [6], however, the increasing inter-domain traffic was neglected.

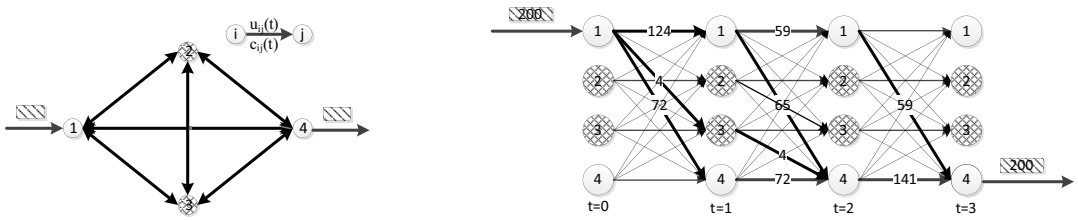
CIPs rely on multiple Internet service providers (ISPs) to provide connectivity of geo-distributed datacenters. Since many economic and business-policy factors affect an individual ISP's decision to peer or not to peer with another ISP, the inter-domain transit capacity is usually limited and ineffective, resulting in poor performance when data transits across multiple ISP domains. According to our experiment, the round trip time (RTT) of inter-domain links is 3-5 times longer than intra-domain links.

Moreover, the growing inter-domain traffic increases the ISP operational cost [7]–[9]. As the authors mentioned in their P4P paper [8], inefficiency in inter-domain traffic may lead to serious disruption to ISP economics. It is not beneficial to reduce the CIPs' bandwidth cost or improve the Quality of Services over the long run. In [9], the authors insisted that inter-domain traffic generated by CIPs should take the ISP operational cost into consideration, that means, to be ISP friendly.

We consider these issues and address the above challenges in this paper. The goal is to design a bulk data transfer strategy that reduces inter-domain traffic as much as possible. In other words, to be ISP friendly. Our approach is named D4D, which means a bulk Data ScheDuling scheme for D2D jobs. Similar to recent studies, we also use time-slotted model [10] to formulate the problem. while unlike to past studies, when we decide which intermediate datacenter to be used, we prefer to the datacenter which is using the same ISP with sending datacenter. By preferentially using intra-domain links , the inter-domain traffic can be significantly decreased.

The rest of the paper is organized as follows. In Section II, we survey related works. In Section III, we describe the system model. In Section IV, we formulate the problem with a

978-0-7695-4807-4/12 $26.00 © 2012 IEEE
DOI 10.1109/CLUSTER.2012.44



(a) Static network (b) Time-expanded network

Fig. 1. An example of time expanded inter-datacenter network

general store-and-forward scheme, and then propose our D4D mechanism. We evaluate our proposed schemes in Section V and conclude in Section VI.

II. Related works

As there exists a large portion of delay tolerant jobs in inter-datacenter communications, many past studies exploited store-and-forward scheme via intermediate datacenters to improve the bandwidth utlization and reduce cost on inter-datacenter traffic.

The authors in [4] discovered that datacenters which were located in different time zones have different leftover bandwidth during same time intervals. They employed a network of storage nodes to stitch together unutilized bandwidth. Based on prediction on the availability of leftover bandwidth at access and backbone links as well as storage constraints at storage relay nodes, they used a store-and-forward algorithm to schedule data transfers and adapt to resource fluctuations.

In [5], Feng *et al.* proposed a similar store-and-forward mechanism. They considered that the inter-datacenter traffic transit costs which are charged by different ISPs varies significantly across different overlay links. They designed a store-and-forward mechanism to minimize costs on inter-datacenter traffic.

The authors in [6] proposed a scheduling algorithm to reduce the peak bandwidth consumed. Different from the method mentioned above, they took both delay tolerant jobs and real-time or critical data into consideration, where the real-time applications use peak bandwidth preferentially. They formulated the problem as a bin packing alike problem.

So far, none of the past studies concerned about the inter-domain traffic. Though similar to [4] and [5], we use a time-expanded graph to model our problem, our target is totally different: To be ISP friendly.

III. System model

In this paper, our goal is to design an inter-datacenter bulk data scheduling strategy which uses the under-utilizated inter-datacenter bandwidth and reduces inter-domain traffic as much as possible. Since the history of bandwidth utilization can be captured at a fine-grained frequency via tools such as Amazon CloudWatch, the leftover bandwidth can be predicted using effective algorithms. We assume the residual bandwidth between datacenters with vary time in a predictable fashion.

In our model, datacenters and the overlay links which interconnect them construct an inter-datacenter network. This network can be modeled as a time-expended graph $G = (N, E, T, u_{ij}(t), c_{ij}(t))$, which is a complete directed graph, where *N* is the set of datacenters, and *E* is the set of overlay links inter-connecting datacenters. Though most of the bulk data is time tolerant, these data still need to be transited in a loose deadline, such as 10 hours, 2 days, etc. We set *T* as the maximum expected transfer time to each bulk data. $u_{ij}(t)$ represents the residual bandwidth capacity of overlay links, which is varying at different time instant. For example, when data is transferred from datacenter *1* to datacenter *2* at time instant t=3, the residual bandwidth between datacenter *1-2* is $u_{12}(3)$. To be ISP friendly, we use $c_{ij}(t)$ to represent preference or we say priority when we choose a relay link. We prefer to choose intermediate links with low $c_{ij}(t)$ value. In practice, $c_{ij}(t)$ can be set by the RTT value between datacenter *i* and datacenter *j* at time instant t. Because intra-domain links usually have lower RTT values than inter-domain links. This function can be easily developed as health check algorithms which have already been implemented in load balancing software, and the RTT values keep stable in several time intervals. Without loss of generality, we consider that each datacenter interconnecting with each other uses single ISP. Even a datacenter uses multiple ISPs, in the graph, we can easily decompose the datacenter into several mirror datacenters. Each mirror datacenter only uses single ISP links.

Our example is shown in Fig. 1, there are *4* datacenters interconnecting with each other by overlay links. Fig. 1(a) is a static inter-datacenter network which reflects that the datacenters are inconnected with each other. We use nodes in different padding to represent different ISPs the datacenter is using. In this figure, we assume the source datacenter is *1*, and the destination datacenter is *4*, which are using the same ISP. The residual bandwidth $u_{ij}(t)$ and the preference $c_{ij}(t)$ are not drawn in the graph because they are time-varying. To make the problem tractable, as in [5], we assume that each bulk data can be transited from the upstream datacenter to downstream datacenter in several time intervals. Fig. 1(b) shows the corresponding time-expanded graph which is constructed from Fig. 1(a) at time instant t=0, t=1, t=2, and t=3. This graph also shows a simple example that a bulk data with volume of *200* units needs *3* time intervals to be transferred from datacenter *1* to datacenter *4*. Each residual bandwidth $u_{ij}(t)$ is generated uniformly random with size of [0,100] units, and the preference $c_{ij}(t)$ of overlay links interconnect datacenters which are using the same ISP are deliberately set lower values

TABLE I
KEY NOTATIONS IN THE SYSTEM MODEL

Symbol	Description
i	i_{th} dataceter in the network
e_{ij}	overlay link between datacenter *i* and datacenter *j*
T	maximum expected transfer time
t	time instant when bulk data start to be transferred
$f_{ij}(t)$	current flow on edeg e_{ij} at time instant *t*
$c_{ij}(t)$	preference of overlay link e_{ij} at start time instant *t*
$u_{ij}(t)$	bandwidth capacity of overlay link e_{ij} at time instant *t*

than those are using different ones. For instance, the $c_{14}(t)$ is usually lower than $c_{13}(t)$. The graph describes that most data is transferred to datacenters using the same ISP. At time instant t=1, even there exists residual bandwidth from datacenter *1* to datacenter *2*, we would rather delay transfer residual data *59* units utill there exists residual bandwidth between datacenter *1* to datacenter *4* at t=2. One reason for transiting *4* units data from datacenter *1* to datacenter *3* at t=0 is that if partial cross-ISP links are not used, it will not be expected to finish transmission in *3* time intervals.

Some key notations used through the paper are summarized in Table I.

IV. PROBLEM FORMULATION

Based on our time-slotted model, the problem of keeping inter-datacenter bulk data transfer ISP friendly while maximizing bandwidth utilization can be formally stated as: when we decide to transmit a bulk data at time instant t, we need to choose which link has adequate capacity to transfer given size bulk data and do not generate inter-domain traffic. It means that we not only need to choose a link which has maximum bandwidth capacity, but also prefer to choose the link interconnecting source and destination datacenters using the same ISP. Since we defined $c_{ij}(t)$ as the preference of overlay link e_{ij}, we can easily formulated the ISP friendly inter-datacenter bulk data transfer problem as a minimum cost flow problem.

A. General store-and-forward Approach

Before presenting our D4D approach, we introduce the general store-and-forward algorithm first. In the general method, each link is viewed as equivalent from the perspective of sending datacenter, the objective is to find a path which has adequate capacity to load the available bulk data. So in a general store-and-forward scheme, the problem can be formulated as equation (1)-(5):

$$min \qquad z = \sum_{t=0}^{T-1} \sum_{e_{ij} \epsilon E} f_{ij}(t) \tag{1}$$

$$s.t. \qquad \sum_{t=0}^{T-1} \sum_{e_{1j} \epsilon E} f_{1j}(t) = F \tag{2}$$

$$\sum_{e_{ij} \epsilon E} f_{ij}(t-1) - \sum_{e_{ji} \epsilon E} f_{ji}(t) = 0 \tag{3}$$

$$\sum_{t=0}^{T-1} \sum_{e_{jn} \epsilon E} f_{jn}(t) = F \tag{4}$$

$$0 \leq f_{ij}(t) \leq u_{ij}(t), \forall \{i,j\} \epsilon E \tag{5}$$

In this formulation, the objective is to find a optimal path to transfer bulk data with volume of *F* units, such that links be used as few as possible to improve efficiency. Equation (2) shows the total sending data from source datacenter equal to the volume of *F*. Equation (3) states the mass balance constraints at all datacenters except source and destination ones. Equation (4) represents the total receiving data at destination datacenter equal to the volume of *F* either. The last inequality (5) shows each flow satisfies the edge capacities.

B. Our D4D Approach

However, as we mentioned before, the above approach does not consider the inter-domain traffic. In order to be ISP friendly and reduce the inter-domain traffic, we only need to add a preference to each overlay links. Since we invariably set the preference of links interconnecting with datacenters using same ISP lower values than those using different ones. Choosing a path with the least total preference value can induce bulk data to be transferred via intermediate links which interconnect datacenters using the same ISP preferentially. As a result, the inter-domain traffic can be reduced dramatically. We use expression (6) to replace (1):

$$min\ z = \sum_{t=0}^{T-1} \sum_{e_{ij} \epsilon E} c_{ij}(t) * f_{ij}(t) \tag{6}$$

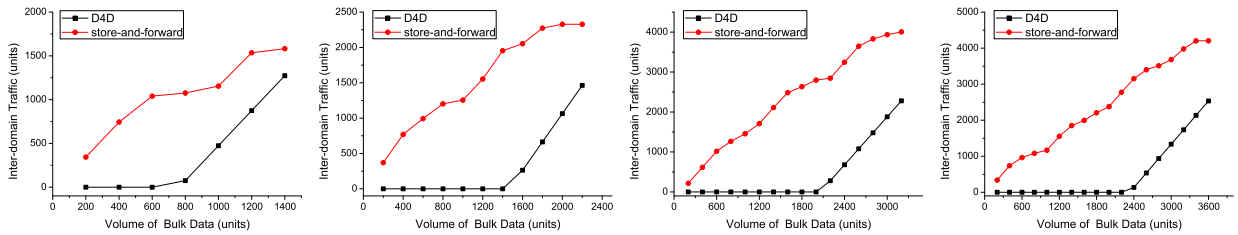
To solve the minimum cost flow problem, we use a classic cycle-canceling algorithm.

V. EVALUATION

We dedicate this section to investigate how D4D performs on reducing inter-domain traffic compared to general store-and-forward algorithm.

The evaluation of D4D is based on our several hundreds lines of codes implemented in C++. We simulate a cloud infrastructure provider with 20 distributed datacenters, in which 12 are using ISP1, and the rest 8 are using ISP2. All the datacenters and overlay links interconnecting them construct a complete directed graph. The residual bandwidth of each link is set to be uniformly random within [1,100] units. The preference of each link is set manually, the preference of overlay links interconnecting datacenters which are using the same ISP are deliberately set lower values than those are using different ones. In our evaluation, the formers are set to be uniformly random within [1,10], and the remainders are set to be uniformly random within [90,100]. The volume of bulk data is set to start from 200 units, with 200 units incremental steps till maximum feasible volume the network can carry. We conduct our simulations 10 times, with maximum expected time intervals T=3, 4, 5, 6 respectively. The evaluation of the general store-and-forward is implemented with the same parameters for comparisons.

We consider four different simulation settings with T=3, 4, 5, 6 respectively, which are shown in Fig. 2(a)-(d). As maximum expected time grows increasingly, there're more nodes shown in the later figures. The reason is that with longer



(a) T=3 (b) T=4 (c) T=5 (d) T=6

Fig. 2. Inter-domain traffic with T=3, 4, 5, 6 respectively

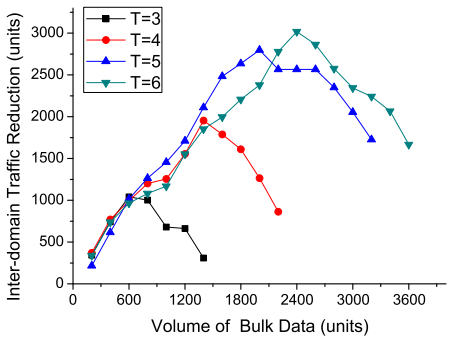


Fig. 3. Inter-domain traffic reduction

tolerant time, the source datacenter can send much more data. The maximum volume of bulk data is gradually increasing correspondingly. Each figure shows that the size of inter-domain traffic generated by general store-and-forward and D4D approaches. The results reveal that our D4D approach outperforms general store-and-forward algorithm significantly. When the volume of bulk data is not very large, the intra-domain links can support all of the inter-datacenter traffic. D4D approach can usually reduce inter-domain traffic to *zero*. By contrast, the general store-and-forward approach chooses links randomly. The inter-domain and intra-domain links are used alternatively, causing inter-domain traffic increases near linearly. However, with the growing of the volume, only using intra-domain links can not transit bulk data in expected time intervals, the inter-domain traffic generated by D4D approach is going to increase. Despite this, our D4D method reduces inter-domain traffic in a considerable volume even when the volume of bulk data reaches the maximum value that the inter-datacenter network can carry.

Fig. 3 summaries the results. In all settings, our D4D approach decreases inter-domain traffic near linearly compared to general store-and-forward approach until the intra-domain links can not carry whole bulk data. With longer tolerate time, data are more possibility to be transited using intra-domain links. Because with larger T, even there do not exists a spare intra-domain link currently, the data can be stored temporarily until there exist a free intra-domain link in a later time. Our D4D approach reaches the ISP-friendliness goal.

VI. Conclusion

This paper describes D4D, an ISP friendly bulk data transfer strategy for geo-distributed datacenters. We first present and formulate the problem in Geo-distributed datacenters. And then we propose an effective scheme named D4D which can reduce the inter-domain traffic significantly. By solving the problem with a time expanded cycle-canceling algorithm, we compared our approach with general store-and-forward approach in several different situation settings. The results revealed that D4D outperformed general store-and-forward approach on reducing inter-domain traffic.

Acknowledgment

The authors would like to thank Jie Zong from the ChinaCache Company, who gives us a lot of relevant suggestions. This work is supported by the National Natural Science Foundation of China(No. 61002011); the 973 Program of China(No. 2009CB320505); the 863 Program of China(No.s 2011AA01A102)

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," EECS Department, University of California, Berkeley, Tech. Rep., 2009.

[2] A. Greenberg, J. Hamilton, D. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 68–73, 2008.

[3] C. Yingying, S. Jain, V. K. Adhikari, Z. Zhi-Li, and X. Kuai, "A first look at inter-data center traffic characteristics via yahoo! datasets," in *Proc. IEEE INFOCOM*, 2011, pp. 1620–1628.

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

[5] F. Yuan, L. Baochun, and L. Bo, "Postcard: Minimizing costs on inter-datacenter traffic with store-and-forward," in *Proc. 2nd International Workshop on Data Center Performance (DCPerf 2012), in conjunction with ICDCS*, 2012.

[6] T. Nandagopal and K. P. N. Puttaswamy, "Lowering inter-datacenter bandwidth costs via bulk data scheduling," in *Proc. IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, 2012.

[7] D. R. Choffnes and F. E. Bustamante, "Taming the torrent: a practical approach to reducing cross-isp traffic in peer-to-peer systems," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 363–374, 2008.

[8] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. G. Liu, and A. Silberschatz, "P4p: provider portal for applications," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 351–362, 2008.

[9] C. Ding, Y. Chen, T. Xu, and X. Fu, "Cloudgps: A scalable and isp-friendly server selection scheme in cloud computing environments," *Proc. 20th IEEE/ACM IWQoS*, 2012.

[10] L. R. Ford Jr and D. R. Fulkerson, "Constructing maximal dynamic flows from static flows," *Operations Research*, pp. 419–433, 1958.