This paper was presented as part of the main technical program at IEEE INFOCOM 2011

A First Look at Inter-Data Center Traffic Characteristics via Yahoo! Datasets

Yingying Chen[1], Sourabh Jain[1], Vijay Kumar Adhikari[1], Zhi-Li Zhang[1], and Kuai Xu[2]

[1]University of Minnesota-Twin Cities
[2]Arizona State University

***Abstract*—Effectively managing multiple data centers and their traffic dynamics pose many challenges to their operators, as little is known about the characteristics of inter-data center (D2D) traffic. In this paper we present a first study of D2D traffic characteristics using the anonymized NetFlow datasets collected at the border routers of five major Yahoo! data centers. Our contributions are mainly two-fold: i) we develop novel heuristics to infer the Yahoo! IP addresses and localize their locations from the anonymized NetFlow datasets, and ii) we study and analyze both D2D and client traffic characteristics and the correlations between these two types of traffic. Our study reveals that Yahoo! uses a hierarchical way of deploying data centers, with several satellite data centers distributed in other countries and backbone data centers distributed in US locations. For Yahoo! US data centers, we separate the client-triggered D2D traffic and background D2D traffic from the aggregate D2D traffic using port based correlation, and study their respective characteristics. Our findings shed light on the interplay of multiple data centers and their traffic dynamics within a large content provider, and provide insights to data center designers and operators as well as researchers.**

***Index Terms*—Content provider, Inter-data center, NetFlow, Anonymization**

I. Introduction

Recent years have seen unprecedented growth in the data center driven technologies and services. Various organizations are now sourcing their computing to "cloud-based" infrastructures. Therefore, large scale data centers and associated cloud services are developed and deployed by various organizations and service providers to store massive amounts of data, and enable "anywhere, anytime" data access as well as computations on the data. Further, for scalability, robustness and performance (e.g., latency), multiple data centers are often deployed to cover large geographical regions. For instance, Microsoft, Google, and Yahoo! own large scale data centers that are located in different geographic locations around the world.

While there are a few recent studies [1], [2] regarding the traffic characteristics within a single data center, little is known about the inter-data center (D2D) traffic dynamics among multiple data centers. Just as the studies of traffic characteristics within a data center, such as workload distribution and where congestion occurs, helps the design and management of data centers, we believe that better understanding of the traffic characteristics between multiple data centers (within a single service provider, e.g., a content provider) and their interactions with client-triggered traffic is critical to effective operations and management of multiple data centers. For instance, such understanding can help in deciding what and how services should be deployed across multiple data centers, what caching and load-balancing strategies [3], [4] should be employed, and how to manage the traffic in the wide-area network backbone connecting the data centers to optimize performance and minimize operational costs [3], [4].

In this paper we present a first study of inter-data center (D2D) traffic characteristics using the anonymized NetFlow datasets collected at the border routers of five major Yahoo! data centers. Our contributions are multi-fold. First, we develop novel heuristics to infer the Yahoo! IP addresses that are involved in data center-client (D2C) traffic and localize their locations from the anonymized NetFlow datasets. Based on several key observations regarding traffic directions and router interfaces, we develop an effective methodology to extract and separate inter-data (D2D) traffic from data center-client (D2C) traffic, and analyze the characteristics of both D2D and D2C traffic and their correlations. Our analysis reveals that Yahoo! organizes data centers in a hierarchical way. In "satellite" data centers, D2D traffic is strongly correlated with the client traffic. In "backbone" data centers, we classify D2D traffic into two categories: i) *client-triggered* D2D traffic, i.e., D2D traffic triggered by the front-end "customer-facing" services such as web search, email, online chat, gaming, video, and so forth; ii) *background* D2D traffic, i.e., D2D traffic due to internal tasks such as routine background computation (e.g., search indexing), periodic data back-up, and so forth. Using novel port based correlation analysis, we are able to further separate these types of D2D traffic, and study their respective characteristics. We find that background D2D traffic has smaller variance, with no significant trends over the day; on the other hand, client-triggered D2D traffic exhibits varying trends over the day. Furthermore, we show that several D2C services are strongly correlated with each other. These correlations among different services have important implications for distributing different services at multiple data centers. For instance, services with highly correlated traffic can be served from the same data center to minimize the inter-data center traffic.

To our best knowledge, our work is the first study of inter-data center traffic characteristics of a large global content provider. It sheds light on the interplay of multiple data centers and their traffic dynamics within a large content provider.

Though the D2D and D2C traffic characteristics studied in the paper may be specific to Yahoo! and the services it provides, our methodology is nonetheless general, and can be applied to understand the D2D and D2C traffic characteristics of any other large content provider or cloud-service provider. All in all, we believe that our work provides useful insight to data center designers and operators as well as researchers.

The remainder of the paper is organized as follows. In Sec. III we provide the overview of the datasets and Yahoo! data centers. Sec. IV presents the methodology for separating Yahoo and non-Yahoo IP addresses, and analysis of inter-data center traffic are presented in Sec. V. Finally, we provide a discussion of the implications for our findings in Sec. VI and conclude the paper in Sec. VII.

II. Related Work

As mentioned earlier, there have been a few recent studies [1], [2] regarding the traffic characteristics within a single data center. In [1], authors provide both macroscopic and a microscopic view of the traffic characteristics and congestion conditions within data center networks. In [2], authors analyze the end-to-end traffic patterns in data center networks, and examine temporal and spatial variations in link loads and losses. On the other hand, little is known about inter-data center traffic characteristics. Similarly in [4], the authors study the YouTube data center traffic dynamics using the Netflow data collected at a tier-1 ISP, with the emphasis on inference of load-balancing strategy used by YouTube and its interaction and impact on the ISP network. Due to the nature of data used, the traffic seen is primarily D2C traffic, and limited to the perspective to a single ISP. To our best knowledge, our work is the first attempt at analyzing and characterizing inter-data center traffic characteristics; we also develop novel methods for separating D2D traffic from D2C traffic, and for further separating background D2D traffic and client-triggered D2D traffic.

III. Overview of Yahoo! datasets

In this section we provide the overview of the Yahoo! data centers and their connectivity. We also describe the network flow datasets [5] used in this study. Further, to facilitate the discussion in the paper we classify the flows into several meaningful categories which is described later in the section.

In this study we consider five major Yahoo! data centers which are located at Dallas (DAX), Washington DC (DCP), Palo Alto (PAO), Hong Kong (HK), and United Kingdom (UK). DAX, DCP and PAO are located in US, and provide most of the core services such as web, email, messenger and games, etc. They are also the largest Yahoo! data centers in terms of the amount of traffic exchanged. At each of the data centers, Yahoo's border routers connect to several other ISPs to reach its clients and other data centers. These data centers are also directly connected to each other through a private network service(e.g. VPN, leased lines etc), and hence may carry traffic for each other through this private network. Fig. 1 provides an overview of the Yahoo! data centers and their connectivity.

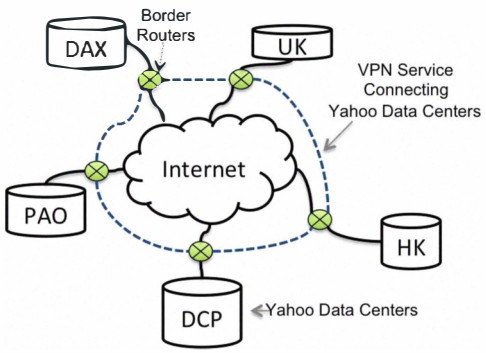

Fig. 1. Overview of five major Yahoo! data centers and their network connectivity.

Our study is based on NetFlow datasets collected at one of the border routers at each of the locations mentioned. Unlike the datasets used in the previous studies related to data center traffic analysis (such as [1], [2]) the NetFlow datasets used in our study provide us with not only the profiling of Yahoo! to "client"[1] traffic, but also the traffic exchanged between different Yahoo! data centers, which we believe is the first such work that sheds light on the inter-data center traffic characteristics for a large content provider. The network flow data collected at each border router, includes both the inbound and outbound traffic. Each record in the NetFlow data contains a "sampled flow" information, which includes following fields: a) *timestamp*, b) source and destination IP addresses and transport layer port numbers, c) source and destination interface on the router, d) IP protocol, e) number of bytes and packets exchanged.

An important challenge with the datasets is that the IP addresses in the network flow traces are permuted to hide the identities of the Yahoo! users. However, prefix-preserving schemes [6], [7] are used in permutation, i.e. if an IP address $a.b.c.d$ is permuted to $w.x.y.z$ then another IP address $a.b.c.\bar{d}$ is mapped to $w.x.y.\bar{z}$. Due to this reason, through out this paper we represent summarized IP address based statistics using /24 IP prefixes. Also, we use the term "client" to represents the non-Yahoo hosts connected to Yahoo! servers. These hosts may be the actual Yahoo! users connecting to Yahoo! servers to access various services, or other servers connecting to Yahoo! servers, such as other mail servers may connect to Yahoo! mail servers to exchange emails.

Classification of Flows: In order to facilitate the discussion in this paper, we classify the flows collected into following two categories:

i. *D2C traffic*: The traffic exchanged between Yahoo! servers and clients.

ii. *D2D traffic*: The traffic exchanged between different Yahoo! servers at different locations.

A border router at a given location may also carry D2C and D2D traffic for other locations. We refer to these types

[1] We refer to non-Yahoo hosts connecting to Yahoo! servers as clients unless specified.

of traffic as *transit D2C traffic* and *transit D2D traffic*, respectively. Accordingly, we also define two types of Yahoo! prefixes. One is the Yahoo! prefixes that are involved in the D2C traffic, referred to as *D2C prefix*. The other is the ones that are involved in D2D traffic, denoted as *D2D prefix*. Note that a Yahoo! prefix can potentially be involved in both D2C and D2D traffic. In fact, we will see in the later sections that there is significant amount of overlap in the prefixes belonging to each category.

IV. Identifying Yahoo! Prefixes

Understanding D2C and D2D traffic characteristics is not possible without identifying the IP addresses used by Yahoo! hosts, and therefore, presents a key challenge to our analysis. In this section we describe our heuristics to infer the IP addresses used by Yahoo! hosts using basic features of the traffic seen at border routers of each data center.

A. Challenges

Inferring original information from anonymized data has already been studied in several other previous studies e.g. [8], [9]. However, these solutions are specific to the datasets, and do not apply for sampled NetFlow datasets. For instance, the inference techniques discussed in [8] require ARP traffic information, hardware addresses in the link layer, as well as other specific header and transport protocol requirements. In addition, they also make use of a lot of other auxiliary public information. Furthermore, authors explicitly note that NetFlow data is invulnerable to their inference techniques because of the lack of required header and transport protocol information. In contrast to the previous work, we need to look at all the services provided by one content provider, with very limited information presented in NetFlow data.

In addition to the limited information provided by the data, there are also several challenges specific to our problem that we need to address. These challenges include the following. 1) Our goal is to study the characteristics of both D2C and D2D traffic. However, the IP addresses involved in each type of communication may have quite different network characteristics, which led to a two-step process in identifying the Yahoo! prefixes. Where, in first step we separate Yahoo! IP addresses from non-Yahoo IP addresses in the D2C traffic, and in the second step we further extract the D2D IP addresses. 2) As we have observed, the border router at one location carries not only its own traffic (i.e. the traffic belonging to one of the hosts at that data center), but also transit traffic for other Yahoo! locations, which does not involve the hosts from the same location. Due to such "transit traffic" carried by Yahoo! border routers for the other Yahoo! locations, Yahoo! prefixes that belong to one location can also appear in the data collected at other Yahoo! locations. Therefore, heuristics to localize the inferred Yahoo! prefixes is needed. 3) Some of the IP addresses used in the D2D traffic may not be announced to other ISPs during the BGP announcements, and therefore it is hard to use the publicly available auxiliary resources, e.g. RouteViews [10], to help inference the data or to validate our inferred results. To address these limitations, we provide novel approaches to inference the NetFlow data. In particular, it is a two-step approach, which consists of identifying the D2C and D2D prefixes, respectively.

B. Identifying Yahoo! D2C Prefixes

We separate Yahoo! prefixes from the client prefixes in D2C traffic based on the degree and ports observed in the flows. A prefix is considered Yahoo! D2C prefix if it talks to large number of other prefixes, and if a large fraction of their traffic uses the TCP ports used by several popular services provided by Yahoo! (such as email, web, messenger etc.). There are two thresholds implied in this heuristic, which are defined as follows. We choose top α prefixes out of all the prefixes based on how many other prefixes these prefixes talk to. Next we choose the prefixes for which at least β fraction of traffic is received at (or sent from) the popular Yahoo! ports. Furthermore, it is important to note that we need to choose the parameters in a relatively conservative manner such that prefixes we get are mostly Yahoo! prefixes, so as to minimize the number of non-Yahoo IP addresses classified as Yahoo! (false negative).

To choose the proper value of β, we first fix $\alpha = 600$, considering top 600 prefixes[2]. In Figure 2(a), red continuous line shows the fraction of traffic for the top 600 prefixes which use Yahoo! service ports, and blue dots represent the fraction of traffic containing Yahoo! ports for the prefixes that each top 600 prefix talks to. Therefore, in this figure, we compare the fraction of traffic that uses Yahoo! service ports on the same side as top 600 prefixes, with the fraction of traffic on the other side of these prefixes. From this figure, we learn that prefixes in the left region, $\beta > 0.5$ are more likely to be Yahoo! prefixes, talking to other prefixes that mostly communicate using popular client ports. In contrast, prefixes in the right region are more likely to be client prefixes. Therefore, we choose $\beta = 0.5$ for DAX.

In order to see how sensitive our D2C prefix inference result is to the change of α value, we experimented with different values of α between 50 to 600, while keeping the value for $\beta = 0.5$. In Fig. 2(b) we show the inference results for three data centers located in US. In this figure, x-axis shows the different values for parameter α and y-axis shows the number of candidate prefixes. We see that candidate prefix set grows initially with the increase in α, however, it becomes stable after α goes beyond 400, and does not increase much by beyond this value. Hence it shows that our D2C inference algorithm is not very sensitive to parameter α, whereby makes easier to find an appropriate value for α.

C. Localizing Inferred D2C Prefixes

The above process only identifies IP addresses (prefixes) that belong to Yahoo!, but could not assign appropriate location to each prefix, due to the challenges mentioned earlier in the section. To assign a correct location to each prefix, we

[2]Using routeviews [10] we found that the number of /24 prefixes announced by Yahoo! ASes at different location is in the range 50-500

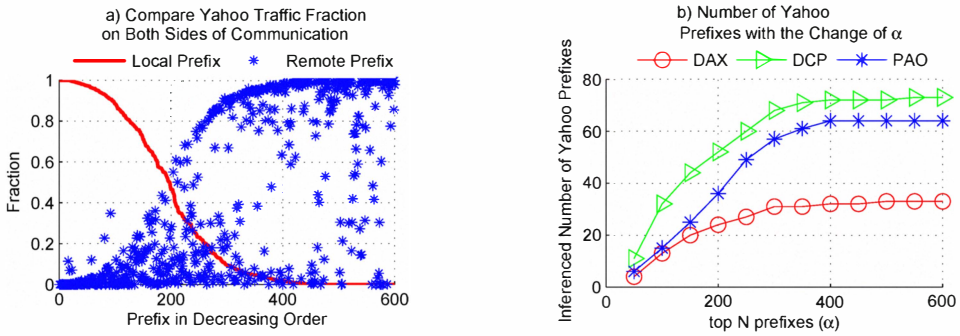

Fig. 2. Choosing parameters α and β.

utilize the traffic direction observed due to the use of early-exit routing, which is prevalent in the Internet [11], [12]. Because of the early-exit routing, the D2C traffic sent from a client and destined to a host at any given data center may enter in the Yahoo!'s network from any border router at another location, and carried through Yahoo!'s own private network. In contrast, the D2C traffic sent from a Yahoo! host to the client always exits Yahoo! network from the same location, and therefore is not carried through the Yahoo! network connecting different locations. We use this observation to locate a Yahoo! IP prefix to its correct data center location. Finally, we assign a location to a Yahoo! IP prefix only if it appears in both incoming and outgoing D2C traffic seen at that location.

D. Identifying Yahoo! D2D Prefix

The heuristics discussed so far are only applicable in identifying the D2C prefixes, however, these heuristics can not extract all the D2D prefixes. It is because prefixes in D2D traffic only talk to a limited number of other Yahoo! prefixes, and the ports used by them may not be listed in the well-known Yahoo! service ports. In addition, unlike asymmetric routing observed in D2C traffic, D2D traffic is mostly symmetric, and carried in Yahoo!'s private network. To infer the D2D prefixes, our heuristics are based on the key observation that there are two types of physical interfaces that play specific roles on each border router.

a. *Foreign interfaces*: All the traffic (including D2D and transit D2C traffic) sends to (or receives from) other data-centers are exchanged through these interfaces on the local border router.

b. *Local interfaces*: These interfaces are only connected to the local hosts at each location.

Since different data centers exchange traffic only through foreign interfaces, a Yahoo! D2D prefix must appear in the traffic that is exchanged through these interfaces. Moreover, to further exclude the possible transit D2C traffic that is also exchanged through the same set of interfaces, a prefix is considered Yahoo! D2D prefix only if its traffic is also symmetric, i.e. both the incoming and outgoing traffic are exchanged through these interfaces. Finally, the local interfaces further help us in completing the list of Yahoo! prefixes at each location.

TABLE I
INFERENCE RESULT.

	D2D prefix	D2C prefix	overlap prefix	D2D IP	D2C IP	overlap IP
DAX	104	108	104	8927	8056	5974
DCP	451	556	446	25299	22020	14257
PAO	280	289	277	15415	12972	7974
UKL	34	35	34	2800	3361	2278
HKX	51	57	51	2226	4795	1754

TABLE II
COMPARING VALIDATED RESULTS AND INFERENCE RESULTS.

	DAX	DCP	PAO	HK	UK
inferred	108	561	292	57	35
IP addresses from local interfaces	106	472	271	20	34

E. Inference Results & Validation

Inference Result: Using the heuristics proposed in this section the inferred prefixes (and IP addresses) are summarized in Table I. It shows the number of prefixes/IPs participating in the D2C traffic and D2D traffic, and the number of overlapping prefixes/IPs in both categories. As we can see, most of the D2D and D2C prefixes overlap. Moreover, the three US locations have more D2D IP addresses than D2C IP addresses, while UK and HK have more D2C IP addresses, implying that more IP addresses are involved in background D2D traffic in the three main data centers in US.

Validation: We validate our results by using testing against some basic constraints. As discussed before, each location have the local interfaces that only connect to the local Yahoo! data centers. Therefore, we first get all the possible local interfaces using our inference results, and see if the union of all the prefixes appearing on these interfaces are close to the number of prefixes we have inferred for each location. If our inferences are not correct, then there is a good chance that we will get a much smaller set of prefixes than extracted by our inference mechanism. Using this validation mechanism we summarize the resulting number of inferred prefixes we get for each location and the union of all the IP addresses appearing at the local interfaces in Table II.

In addition, we also talked to operators at Yahoo! to verify the correctness and completeness of our inference results.

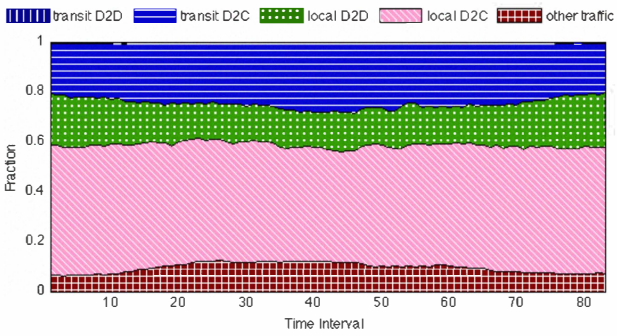

Fig. 3. The fraction of different types of traffic at DAX data center.

Among all the Yahoo! prefixes, our heuristic based inference methodology extracted around 95% (DAX), 95% (DCP), 75% (PAO), 100% (UK), and 75% (HK) of the total prefixes for each location. Further, only less than 5% non-Yahoo! prefixes were classified as Yahoo! (i.e. false negative) and around 5% Yahoo! prefixes were assigned incorrect location. Most of our inference results seem correct, except that we get more than the number of prefixes HK owns. It is not because of the failure of our algorithm, but that HK also carries some traffic from other Asian countries. So these prefixes are coming from other (small) Asian Yahoo! location. However, it will not have negative impact on our D2C and D2D traffic analysis, because these prefixes have been validated to be Yahoo!'s prefixes (i.e. false positive).

V. Traffic Characteristics

In this section we present various characteristics of the traffic seen at the border routers using the inferred D2C and D2D IP prefixes of Yahoo! hosts. In the following we begin with the aggregate statistics for Yahoo! traffic. Next, we present detailed characteristics of D2C and D2D traffic, respectively. In addition, we also present the results on the interaction of D2C and D2D traffic using the port based traffic correlation.

A. Aggregate Traffic Statistics

As described in Sec. III, we classify the traffic seen at the border routers into two categories: i) D2C traffic, and ii) D2D traffic. We further divide each category into two sub-categories, depending upon whether it is destined to the local data center or it is transit traffic seen at that location. The fraction of each type of traffic seen at the DAX data center is described in Figure 3. As seen in this figure more than 50% of the traffic is local D2C traffic at DAX, 20% of the traffic is local D2D, while transit D2C traffic contributes to 25% of overall traffic at DAX. Moreover, a very small amount of traffic is transit D2D. It shows that a significant amount of the D2C traffic seen at the DAX location belongs to the transit D2C, which is expected to be as small as possible. Furthermore, we are not able to classify the remaining 10% of the traffic. It is due to the fact that we define client as all the IP addresses outside these five locations. Since there can not be any client to client traffic, this traffic must be transit D2C and D2D traffic destined to other Yahoo! locations.

TABLE III
Seven categories of D2C services.

D2C service	Port numbers
Email	110, 995, 465, 143, 587
SMTP	25
DNS	53
Messenger	5000, 5001, 5100, 5050, 5061
News	119
Video	1935
Game	11999
Web	80, 443

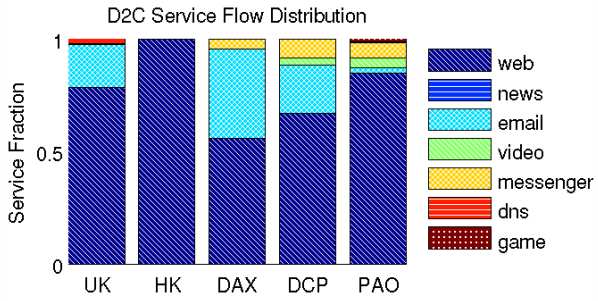

Fig. 4. The distribution of D2C services in each location.

B. D2C Traffic

Yahoo! provides multiple services including email, web-portal, instant messaging, news, music and video. These services are distributed across different data-centers, where each data-center does not necessarily serves all the services. Furthermore, different types of services are also likely to interact with each other. Some services are likely to be correlated with each other, while others may be independent of others. In the following we describe the traffic characteristics for each category of services, and the correlation among them.

1) D2C Service Classification: We identify the Yahoo! services by using the transport layer ports used in the traffic[3]. There are 17 popular server ports observed in our data, which contribute to more than 95% of the aggregate D2C traffic. As we see in Table III most of these ports are well-known such as web and email, while a few of them are specific to services provided by Yahoo! e.g. Yahoo! messenger and video ports. The ports which do not belong to well-known services are identified using entropy of the ports they talk to (see Sec. V-C for details), as well as from the publicly available sources [13], [14]. These 17 ports mainly fall into 7 service categories. The mapping of each service category and the corresponding ports providing this service is listed in Table III.

In Figure 4 we compare the fraction of traffic belonging to each D2C service for all the five data centers. As seen

[3]We consider port numbers for this classification, as no additional information such as application headers, packet payload, etc. is available to us. Nevertheless, it provides a coarser level classification of services and sufficient for understanding general characteristics of various services.

TABLE IV
THE NUMBER OF IPS PROVIDING EACH D2C SERVICE AND THE OVERLAPPING NUMBER OF IPS BETWEEN EACH PAIR OF SERVICES.

	email	DNS	IM	news	video	game	web	SMTP	unique
email	83	8	2	3	1	0	62	67	10
DNS	8	131	2	2	1	0	27	22	102
IM	2	2	235	60	1	1	163	64	71
news	3	2	60	66	0	0	64	64	2
video	1	1	1	0	87	0	67	2	20
game	0	0	1	0	0	2	1	0	1
web	62	27	163	64	67	1	3773	262	3333
SMTP	67	22	64	64	2	0	262	699	424

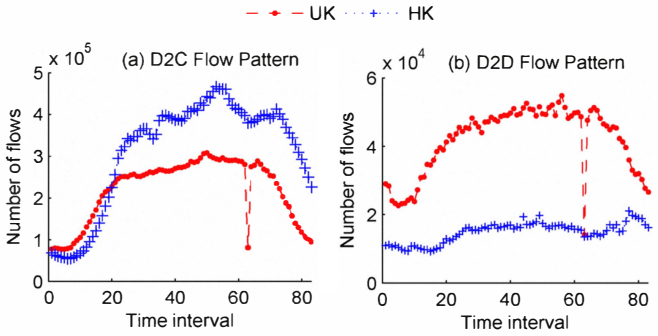

Fig. 6. The D2C and D2D flow patterns during one day in UK and HK.

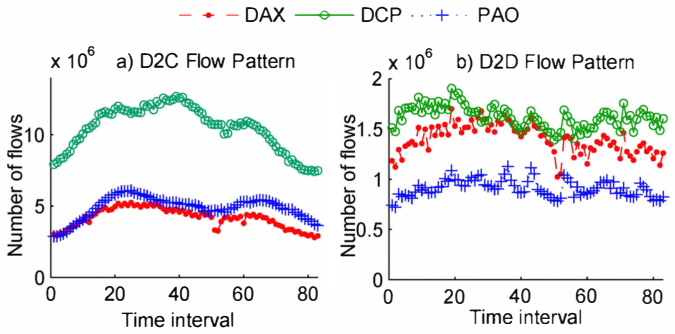

Fig. 7. The D2C and D2D flow patterns during one day in US locations.

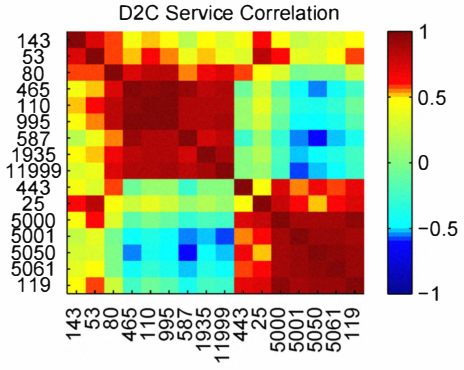

Fig. 5. Cross-correlation between each pair of D2C services.

in this figure, the aggregate D2C traffic is mainly dominated by the web services, which is not surprising as most of the services provided by Yahoo! have web-based interface, and these services are provided at all five locations. On the other hand, instant-messaging (IM), video, and game services have smaller but significant contribution to D2C traffic at all three US locations. Moreover, the choice of location for different services can be affected by many factors such as regional demand, cost of infrastructure and the nature of service itself. Also location based services replicate content at multiple data centers to provide better performance [15], [16]. Table IV shows the number of IPs providing each type of service in DCP data center. We separate port 25 (SMTP) from rest of the email category due to the fact that this is mainly used between Yahoo! mail servers, or between Yahoo! and other service providers' mail servers such as Gmail or Hotmail. On the other hand, other email port numbers are used by clients to directly interact with Yahoo!. The diagonal entries in the table show the number of IPs providing each service as specified in row or column, and the non-diagonal entries show the number of overlapping IPs between two services as specified per row and column. In the last column, we also list the number of unique IPs providing each D2C service. As seen in this table, some of the IP addresses only provide one type of service (see the "unique" column), a large number of them provide multiple services on the same server IP address. From the table we learn that many of the web, SMTP, and DNS services are mostly served using a dedicated set of IP addresses, while the remaining services share IP addresses with other services[4].

2) Cross-Correlation among D2C Services: Though D2C services can be categorized into 7 groups, we find that some of them are strongly correlated (positively or negatively) with each other, while others are independent of each other. We compute the pair wise temporal correlation of each service category to get a better understanding of the interplay among different types of D2C services. Figure 5 shows the correlation between each pair of D2C services in the PAO data center. In this figure, both x-axis and y-axis represent the list of D2C server ports observed in this location. The colored cell corresponding to a pair of services as specified in x-axis and y-axis shows the correlation between them. It turns out that the D2C service ports are clustered into 2 major traffic patterns. The first group consists of several email related ports, and the other messenger ports. These correlations among different

[4]It can happen due to a variety of reasons, such as a single host machine might be running multiple different server instances or a NAT based forwarding is used to divide the traffic to multiple physical(or virtual) servers. It is also likely that these IP addresses are simply frontend servers.

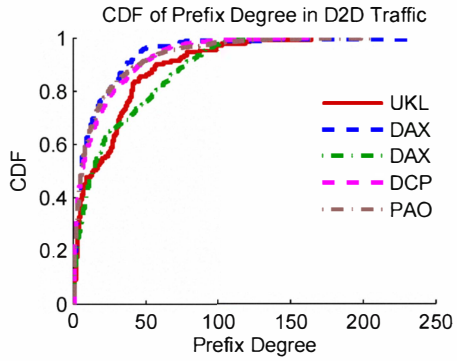

Fig. 8. The prefix degree distribution.

services have important implications for the data center service providers (Yahoo! in this case) to distribute different services at multiple data centers. For instance, services with highly correlated traffic can be served from the same data center to minimize the inter-data center traffic. Further, knowing these correlations information may help them apply more efficient load balancing strategies, and therefore make better use of their computing resources.

C. D2D Traffic

In this subsection, we will first describe the frequency and entropy based technique to identify the popular server ports used in the D2D communication. Next, we describe the D2D traffic characteristics, and its correlation with the D2C traffic.

1) Identifying D2D port: Unlike most of the D2C ports, not all D2D ports are well-known or publicly available. However, the D2C and D2D traffic are exchanged in a similar fashion, namely, following the client/server communication paradigm. That is, in each flow one end-point uses a server port and the other uses a client port. Based on this observation, a port p is considered D2D port only if it meets two constraints. First, p is frequently used in D2D traffic. Second, entropy for the distribution of other ports it talks to is close to 1. We consider top N (in our case, 1000) frequent ports p talks to, and compute the entropy based on the frequency (P_i) of each port appearing on the other end of the flows for p. If it is close to 1, then it is considered as a server port used in D2D traffic, talking to a number of random client ports. By imposing these two constraints, we have found 37 such D2D ports, which cover more than 95% of the overall D2D traffic. Among the 37 ports, the top frequently used ports include 80, 25, 1971, 14011, 5017, 5019, 14020, and 14030.

2) D2D Communication Patterns: To study the aggregate communication patterns among D2D prefixes, we look at the degree distribution for each Yahoo! prefix seen in the D2D communication. Here, we define the degree of each prefix as the number of unique IP prefixes that it talks to. This can be useful in simulating various D2D traffic workloads, to evaluate the network performance. Figure 8 plots the cdf of the prefix-level degree distribution for the each location. As seen in this figure, the prefix-level degree distribution in D2D traffic follows a power-law distribution. Moreover, we observe that each D2D prefix mostly talks to the same set of D2D prefixes in other locations using the same set of D2D ports in our one-day data. This implies that communication patterns among D2D prefixes are quite stable.

TABLE V
CORRELATION COEFFICIENT BETWEEN D2D AND D2C TRAFFIC.

	DAX	DCP	PAO	HK	UK
Correlation	0.81	0.11	0.65	0.87	0.97

3) Cross-Correlation between D2C & D2D Traffic: Figures 6, 7 show the distribution of aggregate D2C and D2D traffic seen in each of the data center locations over time. The x-axis here shows the series of time intervals (15 min for each time interval) during one day. There are 96 15-minute intervals in a whole day. However, the first three and a quarter hour of network data was lost during the collection. Therefore we only show 83 intervals in our analysis. The y-axis shows the number of flows seen in a given interval. The correlation coefficient between the two types of traffic is shown in Table V. When compared with our inference results listed in Table I, we see that D2C and D2D traffic are highly correlated at HK and UK data centers. On the other hand, they are less correlated at the DAX data center, and the PAO data center has only mild correlation, while there is no correlation between D2C and D2D traffic at the DCP data center. The larger the scale of the data center, the less correlated between the D2D and D2C traffic. Interestingly, most of the Yahoo! IP addresses seen at HK and UK data centers appear in both D2C and D2D traffic, which explains the strong positive correlation, as discussed in Sec IV-E. These act more like the "satellite" data centers in the sense that they have smaller scale and the D2D traffic is mostly triggered by the D2C demands. On the other hand, for the three US locations, the D2C traffic has shown varying trends at different times of the day, while D2D traffic does not show any dominant trends. Moreover, we observe that data centers in US locations carry transit traffic for the UK and HK locations, as well as among themselves. In contrast, we do not see any transit traffic in UK, and only a little in HK.

The data centers in US seem to act more like a "backbone" data centers. As we have already seen in Sec IV-E, there are more IPs involved in the D2D traffic in these data centers. Intuitively, D2D traffic in the US locations may be affected by many factors. For example, it can be affected by both the D2C traffic in that location, and the D2C traffic in other locations. There may also exist some background traffic, e.g. regular maintenance or content replication, which might be independent of the D2C traffic. Based on the underlying causes of D2D traffic, we define two major types of D2D traffic:

a. *D2C-triggered* D2D traffic, which is triggered by D2C traffic. If it is triggered by the local D2C traffic, it is defined as local D2C-triggered D2D traffic. If it is triggered by the D2C traffic in other locations, it is foreign D2C-triggered D2D traffic.

b. *Background* D2D traffic, which includes the regular traffic

TABLE VI
THE NORMALIZED STANDARD DEVIATION FOR D2C-TRIGGERED D2D AND BACKGROUND D2D TRAFFIC.

	DAX	DCP	PAO
D2C-triggered D2D traffic	0.1429	0.0887	0.1427
background D2D traffic	0.0994	0.0761	0.0897
$\frac{D2C-triggeredD2D}{backgroundD2D}$	1.4373	1.1669	1.5903

exchanged among the back-end servers, and the traffic incurred by other network events, such as network failure, etc.

The difference between the two sub-types of D2C-triggered D2D traffic is that the local D2C-triggered traffic will actively generate request traffic from a local host to a remote host, i.e., the remote Yahoo! host uses D2D server ports. In contrast, foreign D2C-triggered traffic will trigger D2D traffic that is requested by a Yahoo! server from other data centers, implying that local Yahoo! host uses D2D server ports in the data exchange. We extract the D2D traffic that is triggered by (both local and foreign) D2C, via correlating the D2D traffic at each D2D port with D2C traffic at different ports in each location. The D2D traffic that uses the set of D2D ports that are highly correlated with the D2C ports are considered as the local or foreign D2C-triggered D2D traffic. The D2D traffic that does not use any of the ports that are highly correlated with the local and foreign D2C traffic, is considered as the background D2D traffic.

Our findings show that D2C services are only correlated with certain specific D2D ports. Furthermore, most of the D2C services that are highly correlated with the D2D ports are email-related services. This is quite reasonable, as the email service usually requires a lot of data stored at the back-end data center servers. While for services such as messenger and game, they do not need such supporting data from the back-end servers.

Finally, we extract the background D2D traffic by excluding the local D2C-triggered D2D traffic as well as the foreign D2C-triggered D2D traffic from the aggregate D2D traffic seen at each location. In Figure 9, we compare the background D2D traffic with the two types of D2C triggered traffic. It shows that the background D2D traffic is dominant in the aggregate D2D traffic. Moreover, D2C triggered traffic has increasing or decreasing trends depending upon the time of the day. On the other hand, background D2D traffic does not have any significant trends over the day, but it has smaller variance compared with the D2C triggered traffic. To quantify the variance of the two types of D2D traffic, we use the metric of normalized standard deviation, which normalizes the standard deviation by the mean value of the flow. The results are summarized in Table VI. As seen in this table D2C-triggered D2D traffic has larger normalized standard deviation than background D2D traffic, which implies more stable behavior for background D2D traffic over time.

VI. DISCUSSION AND IMPLICATION

Our findings in the paper have important implications not only to network researchers, but also to the network operators or data center designers. In this section we discuss the various findings made by our study, and their various implications.

Data Inference: There are very limited number of publicly available datasets to understand the inter-data center traffic characteristics. However, most of these datasets are anonymized due to various concerns related to privacy of users, security of data center infrastructure etc. These obstacles severely limit the usefulness of these datasets. To overcome these challenges, we developed some simple and intuitive heuristics, which proves to work far better in terms of accuracy than some complicated ones, such as correlating the timestamp between different flows etc., which is commonly used in traffic analysis and correlation [17]. Because of its simplicity, the alogrithms can be easily adjusted or directly applied to any other anonymized datasets from other content providers.

Flow Classification: Since most of the existing work related to network traffic analysis focuses on single data center, and content providers are usually not willing to publicize their data center locations and internal topology, little is known about the types of traffic we might see among different data centers within one content provider. Our study shows the presence of various types of traffic, such as client to server traffic, traffic among servers at different data centers, etc. However, it is a challenging task to separate these flows from the aggregate network traffic. Using the correlation based techiques developed in this paper, we have provided an initial estimate of such traffic and their characteristics.

Traffic Correlation: In general, data centers are used to provide various services with different characteristics. Due to the co-existence of several services, it becomes difficult to understand how the traffic for different services interact with each other. On the other hand, a better understanding of these interaction can help in developing better strategies to deploy various services across data centers, to optimize their network performance. For instance, D2C services with highly correlated traffic can be served from the same data center to minimize the inter-data center traffic, which has shown to be quite large in Yahoo!. Moreover, by correlating D2D and D2C traffic, we infer that Yahoo! uses a tiered structure in deploying their data centers, with several "satellite" data centers mostly distributing services, and "backbone" data centers having huge amount of background D2D traffic going on. By inferring and extracting background D2D traffic, we are able to estimate how much background traffic may exist within a content provider. By analyzing its characteristics, we show that background D2D traffic exhibits quite irregular, often varying, patterns and trends. These characteristics have important implications for data center operators or designers, and can help them in designing efficient schemes for deploying/managing data centers, doing content replication, as well as a lot of other background operations.

VII. CONCLUSION

Understanding data center traffic dynamics is critical for designing and managing large scale data centers. Besides a few recent studies [1], [2] of traffic within a single data center,

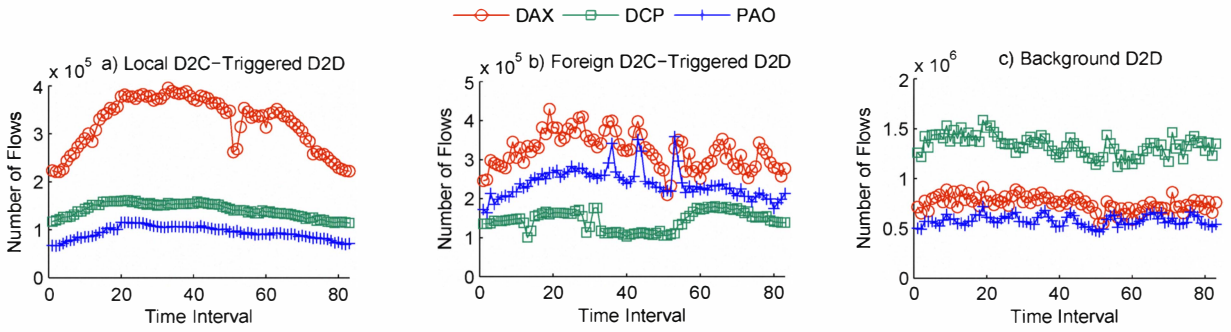

Fig. 9. Comparing three types of D2D traffic.

little is known about inter-data center traffic dynamics. In this paper, using the network traces collected at five major Yahoo! data centers, we provided the first study on traffic dynamics among multiple data centers within a large global content provider. Our results indicated that Yahoo! employs a hierarchical way of deploying its data centers. In "satellite" data centers, D2D traffic is closely correlated with D2C traffic. For the three US locations, we identified two types of D2D traffic: i) D2C triggered traffic and ii) Background D2D traffic. By applying port based correlation, we separated these types of D2D traffic. Our findings showed that background D2D traffic is quite dominant in the aggregate D2D traffic. At the same time, it shows no significant trends, and has smaller variance compared with the D2C triggered traffic. On the other hand D2C triggered traffic shows varying trends over the day which are mainly governed by the user dynamics, and has larger traffic variance. Also, generally these data centers provide multiple services which may be located(and replicated) at different data centers. We also showed that several of Yahoo! services have correlated traffic. These correlations have important implications for distributing different services at multiple data centers. In addition, we also developed simple traffic feature based inference techniques to separate the Yahoo! and non-Yahoo! IP addresses using the anonymized NetFlow traces. The proposed techniques not only perform really well on the Yahoo! NetFlow datasets, it is simple, intuitive, and general, therefore can be applied to anonymized NetFlow traces of other providers as well.

ACKNOWLEDGMENTS

The work is supported in part by the NSF grants CNS-0721510, CNS-0905037, CNS-1017647 and CNS-1017092. We would also like to thank Yahoo! Inc. for providing us with the datasets.

REFERENCES

[1] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of data center traffic: measurements & analysis," in *IMC '09: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*. New York, NY, USA: ACM, 2009, pp. 202–208.

[2] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding data center traffic characteristics," in *WREN '09: Proceedings of the 1st ACM workshop on Research on enterprise networking*. New York, NY, USA: ACM, 2009, pp. 65–72.

[3] R. Krishnan, H. V. Madhyastha, S. Srinivasan, S. Jain, A. Krishnamurthy, T. Anderson, and J. Gao, "Moving beyond end-to-end path information to optimize cdn performance," in *IMC '09: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*. New York, NY, USA: ACM, 2009, pp. 190–201.

[4] V. K. Adhikari, S. Jain, and Z.-L. Zhang, "Youtube traffic dynamics and its interplay with a tier-1 isp: An isp perspectives," in *IMC '10: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement conference*. New York, NY, USA: ACM, 2010.

[5] "Yahoo! research webscope program," http://labs.yahoo.com/Academic_Relations.

[6] T. Brekne, A. Årnes, and A. Øslebø, "Anonymization of IP Traffic Monitoring Data: Attacks on Two Prefix-Preserving Anonymization Schemes and Some Proposed Remedies," in *Privacy Enhancing Technologies*. Springer, 2005, pp. 179–196.

[7] J. Fan, J. Xu, M. Ammar, and S. Moon, "Prefix-preserving IP address anonymization: Measurement-based security evaluation and a new cryptography-based scheme," *Computer Networks*, vol. 46, no. 2, pp. 253–272, 2004.

[8] S. Coull, C. Wright, F. Monrose, M. Collins, M. Reiter *et al.*, "Playing devil's advocate: Inferring sensitive information from anonymized network traces," in *Proceedings of the Network and Distributed System Security Symposium*. Citeseer, 2007, pp. 35–47.

[9] S. Coull, M. Collins, C. Wright, F. Monrose, M. Reiter *et al.*, "On web browsing privacy in anonymized netflows," in *Proceedings of the 16th USENIX Security Symposium*, 2007, pp. 339–352.

[10] Routeviews, "University of oregon route views archive project," *http://archive.routeviews.org*, 2010.

[11] J. Rexford, "Route optimization in IP networks," *Handbook of Optimization in Telecommunications*, pp. 679–700, 2005.

[12] N. Spring, R. Mahajan, and T. Anderson, "The causes of path inflation," in *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2003, pp. 113–124.

[13] "List of tcp and udp port numbers," http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers.

[14] "TCP/IP Ports," in *url http://www.chebucto.ns.ca/~rakerman/port-table.html*.

[15] R. Krishnan, H. V. Madhyastha, S. Srinivasan, S. Jain, A. Krishnamurthy, T. Anderson, and J. Gao, "Moving beyond end-to-end path information to optimize cdn performance," in *IMC '09*. New York, NY, USA: ACM, 2009, pp. 190–201.

[16] C. Huang, A. Wang, J. Li, and K. W. Ross, "Measuring and evaluating large-scale cdns (paper withdrawn)," in *IMC '08*. New York, NY, USA: ACM, 2008, pp. 15–29.

[17] P. Bahl, R. Chandra, A. Greenberg, S. Kandula, D. Maltz, and M. Zhang, "Towards highly reliable enterprise network services via inference of multi-level dependencies," in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2007, p. 24.