Characterizing Traffic Footprint of a Stub-AS

Bahador Yeganeh
Computer & Information Science Department
University of Oregon
Email: byeganeh@cs.uoregon.edu

***Abstract*—A rich and very dynamic Content distribution ecosystem enables today's Internet users to access content associated with major providers from a close-by front-end server. While the advantages of the resulting locality of traffic for users (i.e. better performance) and providers (i.e. lower cost) are well understood, two basic questions about the content distribution ecosystem remained unanswered *(i)* What is the typical level of traffic locality for users at the edge of the network? and *(ii)* Whether the traffic locality affects the observed bandwidth by these users?**

This paper presents an empirical assessment of traffic locality for a stub-AS to answer these questions. We employ unsampled Netflow data from University of Oregon's (UONet) border gateways to assess the level of locality for top content providers of this network using four metrics of distance, namely geographic distance, router hop, AS hop and round-trip time (RTT). We also present a method to identify front-end servers of a major content provider, namely Akamai, that are placed within other ASes with the intention to increase their traffic locality (called guest servers). Using this method, we identify Akamais guest servers that serve UONet. Finally, we examine the effect of traffic locality on the observed performance (i.e. download bandwidth) by UOnet users and show that traffic locality does not have a strong correlation with the download bandwidth of UOnet users.

I. Introduction

Over the past years, the Internet has seen a great increase in the number of its users and has been employed as a medium for delivering various types of content ranging from small sized objects such as a text document for a news headline to large files such as operating system updates and content with more constraints such as video streams for covering live events which demand a greater network performance for their fluid functionality. The ecosystem containing content providers (CP's) and consumers includes other players such as Content Distribution Networks (CDN's), stub-ASes and upstream providers. The Internet has recently observed a topology change which has been referred to as the flattening of the Internet by scholars [1], [2]. In this paradigm content providers will move their front end servers closer to the consumer or the edge of network for two reasons. First, to reduce their costs for paying upstream providers for carrying their traffic. Second, by decreasing their distance to the customer they could decrease their delay and packet loss which would result in higher performance for the customer, this in turn would result in a better user experience which has been linked to more revenue [3].

To distribute their content, CP's have employed various tactics to deliver their content from servers which are more *local* to the client. Google has expanded its presence by placing cache servers in other ASes and would redirect users to these caches using EDNS capabilities [4]. Akamai has placed many thousands of servers across the globe to form its CDN network, Akamai employs it's own algorithms to cache content based on their popularity on its most local servers [5]. Other CP's have relied on the infrastructure that is available through cloud service providers and would often rely on the cloud providers load balancing services to offload their traffic through various datacenter locations [6].

While the benefits of traffic locality for the different constituents of the content distribution ecosystem are well understood and various approaches have been employed to achieve it in today's Internet two questions remain unanswered: *(i) For a typical stub-AS to what degree do we observe traffic locality for its clients. (ii) How much does this amount of traffic locality affect the clients performance.* To answer these two questions we rely on the unsampled Netflow dataset for the campus network of University of Oregon called UONet. For our study we focus on incoming flows and map the source IP address to their corresponding AS. We identify major CP's that deliver the bulk of traffic to UONet regarding bytes and number flows this leads to a list of 12 target ASes. In Section VI, to asses traffic locality we introduce four distance measures namely GEO, router hop, AS hop and RTT we study traffic locality at an AS and prefix granularity and conduct a set of live experiments to give insight into the most active prefixes purpose. For example, we observe that about 50% of UONet's traffic is delivered from a 30ms radius. In Section VIII, we introduce a technique to identify servers that are residing in other ASes and refer to them as guest-servers, we apply this technique to Akamai as a case study and compare the effect of these servers on Akamai's locality. We study the implications of locality on user perceived performance with respect to bandwidth and state that we cannot observe a strong correlation between our distance metrics and bandwidth in Section IX.

The remainder of this paper is organized as follows: we present the structure of UONet and outline why it could be considered as a representative for other stub-ASes in Section II. In Section III, we outline our vantage points for data collection in UONet and present some basic meta stats for our collected data. In Section IV, we present our algorithm for defragmenting flows that have been captured by UONet's gateways. The list of snapshots along with the temporal trend that we have observed over the timespan of 2 years is presented in Section V. Section VI outlines our methodology for assessing traffic locality and identifying the

major CP's for UONet. Section VIII explains our methodology for identifying guest servers and presents the results for our case study (Akamai). The implications of traffic locality on UONet clients bandwidth is presented in Section IX. Section X gives an overview of related works and Finally in Section XI we conclude the paper.

II. UONet: An Overview

University of Oregons network (called UONet) provides Internet connectivity to the campus and off campus residential units. A summary of the population stats for University of Oregon and their breakdown along the different sections of the network is presented in Table I. As we can see UONet is serving a large population with a wide demography thereby it could be a good representative of any stub-AS.

Student Population	Staff Population
24,548	4500

TABLE I
General statistics regarding the population of students in University of Oregon

The topology of UONet among the upstream providers is given in Figure 1. UONet has three upstream Internet providers, namely Nero networks, Oregon exchange(OIX) and Oregon gigapop(OGIG). The access to these providers is provided through two border gateways namely UONet9 and UONet10.

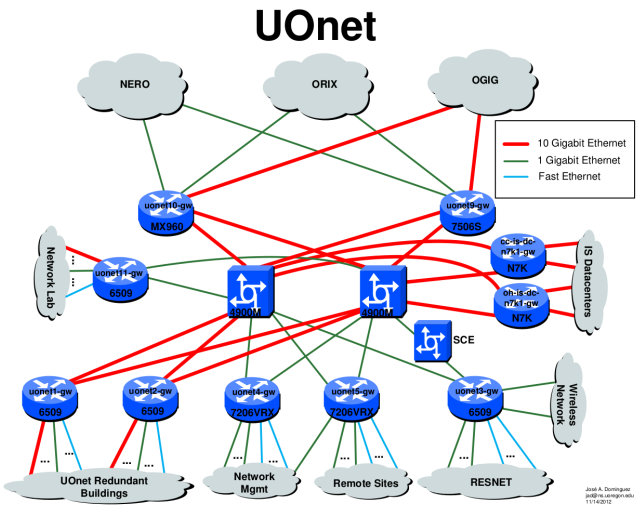

Fig. 1. topology of University of Oregon network, along with the border gateways and the upstream Internet providers.

UONet's traffic is composed of three different sections based on the end point at UONet, namely Wireless, Residential halls(Resnet) and Wired. Wired is associated with users that are connected to UONet through wired caballing and mainly consists of UONet's infrastructure such as web servers, mail servers, etc. Our wireless and residential networks currently only supports IPv4 addresses, while the remainder of traffic that consists of the traffic associated with servers and users that are connected through Ethernet cables supports both IPv4 and IPv6. For a given snapshot the breakdown of UONet's traffic among these three categories is given in Table II.

Section	Bytes %	Flows %
Wireless	58.18	22.89
Residential	16.20	7.75
Wired	25.62	69.36

TABLE II
Breakdown of traffic among different sections of the network for a daily snapshot corresponding to 2015-02-04.

III. Data Collection

The Information Services of University of Oregon has been recording the netflow data at the two border gateways for about two years up to this date. The Netflow data that is being captured at UONet9 is unsampled while UOnet10 is logging data with a sample rate of 1/100. Daily statistics for captured Netflow data at these two border gateways is provided in Table III.

Gateway	Sample Rate	Volume of Traffic	Number of Flows
UONet9	1/1	20.8TB	637M
UONet10	1/100	533GB	17M

TABLE III
Summary of border gateway stats, UONet10 values are projected values based on the reported sample rate. The traffic stats correspond to the 2015-02-04 daily snapshot.

The IP addresses of UONet nodes in captured Netflow data is anonymized by the Information Services in a prefix preserving manner to ensure the privacy of UONet users. The Netflow collector of Information Services stores the files in 5 minute snapshots. On average each 5 minute file is about 50MB and the storage requirement for 1 day is around 14GB. The Information Services employs the **nfcapd** capture daemon as their collector and thereby the files are stored in the nfdump file format. The nfcapd daemon uses fast LZO1X-1 compression on it's output files in order to decrease the storage requirements for the Netflow data [7].

Our data contains the continuous Netflow data of the boarder gateways starting from the beginning of June 2013. We also have the data associated with a subset of the days of February, March, April and May of 2013.

The Netflow data that is being captured at the boarder gateways complies with Netflow v5 and thereby contains the following attributes for each record:

- start, end timestamps
- source, destination IP addresses
- source, destination port numbers
- IP header protocol number
- TOS field of IP header
- TCP flags set during the duration of a flow
- number of packets
- number of bytes
- input, output router port numbers
- source, destination AS numbers

Each bidirectional connection is presented as two separate flows in our records, one corresponding to the outbound packets and another for the incoming packets. One should note that since we have two border gateways, there is a possibility that two sides of a connection would be logged on two different border gateways. Since UONet10 is sampled, we will

not be able to capture all of the traffic that is traversing through that gateway. However, we observe around 16TB & 600GB of exchanged traffic over UONet9 and UONet10 respectively. In short, more than 97% of external UONet traffic goes through the UONet9 gateway. We also examined all the ASes whose corresponding flow are delivered through UONet9 and 10. We noticed that except for one AS the external ASes are mutually exclusive. We believe that captured Netflow traffic at UONet9 provides a representative view of UONet's traffic. The list of top ASes we observed over the UONet10 gateway is given in Table IV, in the remainder of sections the reader would observe that aside for one AS (Level3) the remainder of our target ASes display a symmetric behavior for their point of ingress and egress to UONet. For the analysis in this paper we only use Netflow data from UONet9.

AS#	AS Name
13825	TROYCABLE-NET - Troy Cablevision Inc.
20738	AS20738 Webfusion Internet Solutions
2140	ISSC-AS - IBM Corporation
786	JANET JISC Collections And Janet Limited
3356	LEVEL3 - Level 3 Communications Inc.
28573	NET Servios de Comunicao S.A.
159	OSUNET-AS - The Ohio State University
5408	GR-NET Greek Research and Technology Network S.A
4385	RIT-ASN - Rochester Institute of Technology
23456	23456
4134	CHINANET-BACKBONE No.31Jin-rong Street
680	DFN Verein zur Foerderung eines Deutschen Forschungsnetzes e.V.
15964	CAMNET-AS
557	UMAINE-SYS-AS - University of Maine System
2607	SANET Slovak Academic Network
28260	Altarede de Teresopolis Provedor de Internet Ltda
55	UPENN - University of Pennsylvania
59	WISC-MADISON-AS - University of Wisconsin Madison
1213	HEANET HEAnet Limited
58302	SAMFUNDET-AS Studentersamfundet i Trondhjem
3701	NERONET - Network for Education and Research in Oregon (NERO)
50113	SUPERSERVERSDATACENTER MediaServicePlus Ltd.
24389	GRAMEENPHONE-AS-AP GrameenPhone Ltd.
9121	TTNET Turk Telekomunikasyon Anonim Sirketi
23752	NPTELECOM-NP-AS Nepal Telecommunications Corporation Internet Services
4812	CHINANET-SH-AP China Telecom (Group)
12876	AS12876 ONLINE S.A.S.
1103	SURFNET-NL SURFnet The Netherlands
27768	CO.PA.CO.
210	WEST-NET-WEST - Utah Education Network
6360	UNIVHAWAII - University of Hawaii
11492	CABLEONE - CABLE ONE INC.
35804	ALNET-AS PP SKS-Lugan
9829	BSNL-NIB National Internet Backbone
14048	MEMPHIS-EDU - The University of Memphis
60897	ASIDEAL IDEAL HOSTING SUNUCU INTERNET HIZM. TIC. LTD. STI
23650	CHINANET-JS-AS-AP AS Number for CHINANET jiangsu province backbone
18403	FPT-AS-AP The Corporation for Financing & Promoting Technology
41572	HAFSLUND Kvantel AS
4837	CHINA169-BACKBONE CNCGROUP China169 Backbone
54888	TWITTER-NETWORK - Twitter Inc.
6377	4JNET - Eugene School District 4J
54994	WANGSU-US - MILEWEB INC.
58085	TCE-ASN Esfahan Telecommunication Company (P.J.S.)
3582	UONET - University of Oregon
37110	moztel-as
4600	UO-TRANSIT - Oregon GigaPOP
10876	MAOZ-ASN - MAOZ.COM
766	REDIRIS Entidad Publica Empresarial Red.es

TABLE IV
TOP ASES OBSERVED OVER UONET10 FOR DAILY SNAPSHOT 2015-02-04.

Data Annonymization: To maintain the privacy of UONet users, the Information Services of University of Oregon anonymizes the internal IP addresses of each Netflow record that is associated with University of Oregon clients. The anonymization process is done in a prefix preserving fashion and thereby each UO IP will get mapped to the same IP address therefor we are able to distinguish the flows associated with a single client or computer, as long as the client doesn't get a new IP address from the DHCP server.

To indicate whether the source or destination IP address of a Netflow record is anonymized, the Information Services uses the first two bits of the TOS field. Bit 0 is marked when the destination IP address is anonymized and bit 1 is marked for the source address.

We also performed some validation on the direction of flows based on the router port numbers. We asserted that flows identified as incoming should enter through one of the outbound interfaces and exit through one of the inbound interfaces. A small portion of the flows violated this assertion after taking further look we realized that these flows where spoofed packets with an IP address that matched the anonymization prefixes employed by the Information Services anonymization program and were falsely marked as anonymized. Since the overhead of checking this assertion was high and the fragment of flows that met this condition were negligible (less than a percent), we skipped this step for the remainder of our methodology.

IV. DATA CLEANING & VALIADTION

The algorithm outlined in this section is applied to the flows that have been observed on UONet9 and cannot be applied to UONet10 since the flows captured on this gateway are sampled. For reasons specified in Section III our defragmentation algorithm might present meta flow stats that are less than the real values of the connections they represent. That is if a portion of a flow is handled by UONet10 we will not count the meta statistics such as number of bytes and packets in our final results.

Flow Defragmentation: Once a router observes a new flow over one of its interfaces, it creates a Netflow record for that connection inside its internal cache. Once the connection is terminated, the record is dumped to a Netflow collector which in turn would store the entry to a permanent storage. Since the amount of cache that is available on a router is limited, once the cache is full, the router flushes its cache to free up space for new entries. This would cause a single connection to be split up into multiple Netflow records on the collectors side. In order to perform correct flow level analysis over our data, we need to defragment the Netflow entries that belonged to a single connection.

In order to read Netflow records, one option is to take advantage of the nfdump program that extracts flow attributes and also has the capability to report aggregate statistics on the specified files. One shortcoming of the nfdump program is its inability to defragment flows that were fragmented by the router. For this reason, we needed to develop a code that would read the records and would stitch records of the fragmented flows back into a single record. We could have used the nfdump program to capture the flow attributes and later on use those values to defragment the flows. However, this approach would require two passes for data processing over our files which induces additional processing time and requires additional storage. Thereby, we developed our optimized processing code that employs the nfread library [8] to read the netflow records from the raw files. The nfread library takes advantage of the source codes of the nfdump

software package and enables C or C++ applications to read the captured records.

Note that we only perform the defragmentation algorithm for TCP flows since the notion of connection for other types of flows is either irrelevant or not clearly defined. The algorithm for identifying and defragmenting flows into a single record is as follows: The program will start to read records from the file one by one and will store these flows inside a map data structure that will sort the flows based on the signature of a flow which consists of:

- source & destination IP addresses
- protocol number of IP header
- source & destination port numbers

Each flow with a similar signature would be stored in a multiset data structure which sorts its entries based on the start time-stamp of each flow. By employing this data structure we bundle flows with a similar signature with each other and sort these flows by their start time-stamp.

After examining our dataset, we realized that flows could have a start time-stamp that doesn't belong to their associated Netflow file but the end time-stamp always falls within each files time-frame. Although the number of flows with this characteristic is minuscule, it could affect the validity of our defragmentation algorithm. We calculated the time difference between the start of each flow and the start of the corresponding window over a 24 hour worth of data. The distribution of this time difference is shown in Figure 3. As it is evident from the distribution, once we finish processing round N, we could be certain that we have observed all of the flows that have a start time-stamp within the time-frame of round N-2, since the timespan of the flows of each file corresponds to a time-frame belonging to 2 prior files. Figure 2 shows the misalignment between the time-stamp of a flow and the time that we observe it while we are iterating over the Netflow files.

```
IF(record.protocol == TCP) {
  insert records signature into signature map
  insert record into set of records with a similar
      signature
}

\\at the end of file N
FOR every signature in signature map {
  IF(not updated since file N-2) {
    tmpRecord = first record of this signature set
  FOR every record with similar signature {
    tmpRecord.endTimestamp = max(tmpRecord.
        endTimestamp, record.endTimestamp);
    tmpRecord.byteCount += record.byteCount;
    tmpRecord.packetCount += record.packetCount;
    tmpRecord.stitchCount++;
  }
  }
}
```

Listing 1. flow defragmentation algorithm

The pseudo code of the defragmentation algorithm is given in Listing 1. The semantics of our defragmentation algorithm are quite simple, we parse individual Netflow files based on their timestamp and map their flows on a common timeline as shown in Figure 2. After reading the flows associated with file N+2 we would start looking at the flows that belong to file N and update the following values for flows that have a similar signature:

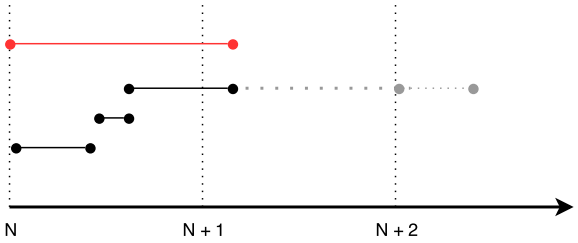

Fig. 2. Misalignment between the time-stamp of the flow and the time-stamp of the file that it has been dumped into. The gray flow shows the window in which we observe the flow while the corresponding black flow shows the time-stamp that the flow has. The red flow indicates the flow after being defragmented.

- end time-stamp
- byte count
- packet count
- stitch count

The distribution of the length of the flows before and after running the defragmentation algorithm are shown in Figure 4. As it is evident from the distribution, defragmented flows have a longer duration and the percentage of long lived flows has increased after running the defragmentation algorithm. On average 25% of the flows are defragmented by our algorithm which is close to findings of [9]. For the 2015-02-04 snapshot, we have 637 and 530 million flows before and after running our defragmentation algorithm, respectively. The distribution for the number of times a flow has been stitched is shown in Figure 5, we observe that among the flows that have been defragmented the majority of them get stitched less than 10 times.

V. Datasets

Due to the large size of our dataset and the huge overhead of processing all of the data, we select the first Wednesday from each month in our dataset for our basic analysis. In addition to these daily snapshots, we use 7 consecutive days to conduct our temporal analysis (*i.e.*weekly snapshot).

Figure 6 lists the selected daily snapshots along with the amount of bytes and flows we observe in each snapshot. The breakdown of traffic along different sections of the network is depicted using different colors. For each snapshot we have two bars one representing the amount of Bytes and the other bar which is shaded presents the number of flows that were delivered to UONet during that snapshot.

As we can see from Figure 6, the volume of traffic for months of July, August and September decreases since most of the students leave campus for the summer break. We can observe that from February of 2014 to 2015 we see about 14% increase in the volume of traffic and about 10% increase in the number of flows. The breakdown of traffic among different

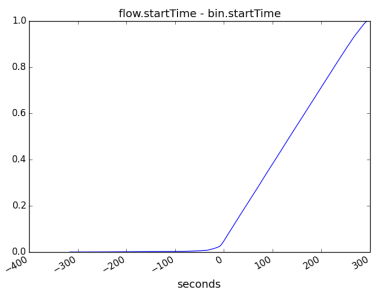

Fig. 3. Distribution of time difference between the start of a flow and the start of its corresponding window over flows of 2015-02-04.

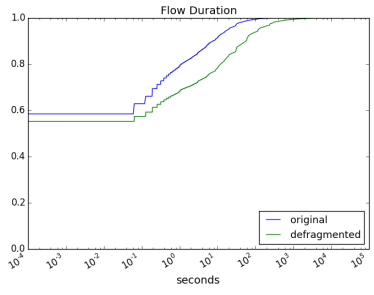

Fig. 4. Distribution of flows length before and after running the defragmentation algorithm.

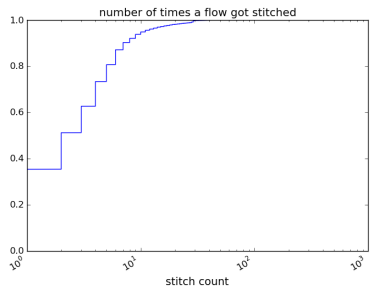

Fig. 5. Distribution of the number of times a flow has been stitched.

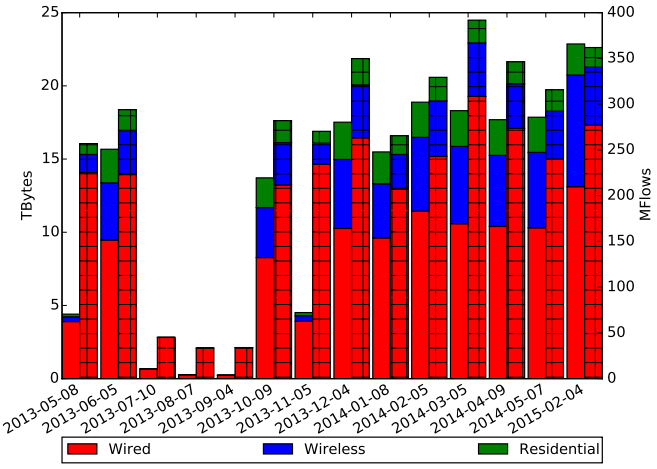

Fig. 6. List of daily snapshots along with the breakdown of traffic stats among different sections of the network.

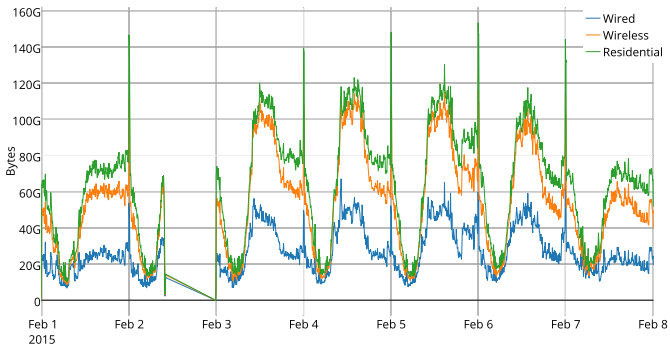

Fig. 7. Breakdown of traffic among different sections of the network for the weekly snapshot.

sections of the network for our weekly snapshot could be viewed in Figure 7.

We observe that the peek of residential and wireless traffic occur at different times of the day, residential traffic increases at night time when students return to their dorms or houses while wireless traffic observes its peek in the midst of the day. We should note that the missing data for Feb 2nd and 3rd is caused by the unavailability of Netflow data during these days.

VI. Top Content Providers

In order to study the delivery mechanism of content providers(CP) to UONet, we focus on all of the incoming flows and group them based on their source AS in each daily Netflow snapshot. To identify the source AS of each flow, we use the Cymru [10] service to map its source IP address to its corresponding AS. The contribution of each AS in the incoming traffic to UONet in each snapshot is measured based on the number of flows and their corresponding bytes. We identified the top 50 ASes with the largest contribution in incoming traffic in each daily snapshot. Table XIII presents this list for the 2015-02-04 snapshot.

Based on the popularity of each AS and users demand the list of top ASes could change from one snapshot to another. The evolution of the top 50 ASes and their prevalence could be viewed as a heatmap in Figure 8. The X axis shows the union of all top ASes we observed over our snapshots while the Y axis shows the rank of the AS within the top 50 list. The color of the cells indicate how many times a specific AS has been observed in that rank. From Figure 8, we can observe that a handful of ASes (the bright blocks) are shown persistently among the top ASes while the remainder of ASes only show up in one snapshot.

Based on the intuition gained from Figure 8 and to limit the scope of our analysis we hand picked 12 ASes for the remainder of our analysis the list of these ASes is presented in Table V. These ASes have a big contribution towards UONet's traffic and are among the well known players in the Internet that either directly or indirectly deliver content for the Alexa [11] top websites. The contribution of these providers towards the traffic of UONet in terms of incoming Bytes and Flows are presented in Figure 9. As we can observe from this figure, these 12 ASes are responsible for 74% of total Bytes and 23% of total flows that were delivered to UONet.

VII. UONet Footprint

The goal of this paper is to study UONet's network footprint on the Internet as a sample Eyeball AS. By footprint we are referring to the GEO and network distance of the content that is being pulled by the users of UONet.

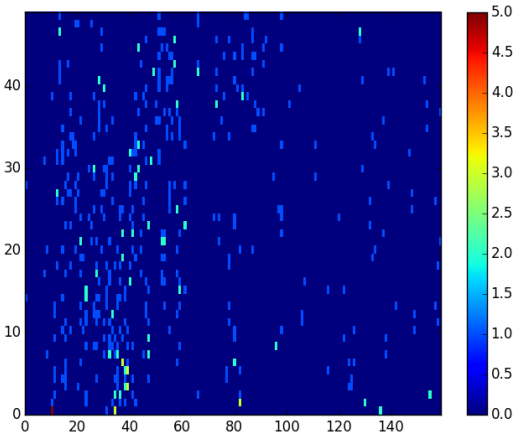

Fig. 8. Prevalence of top ASes among all of the daily snapshots. The Y axis the rank position of the AS.

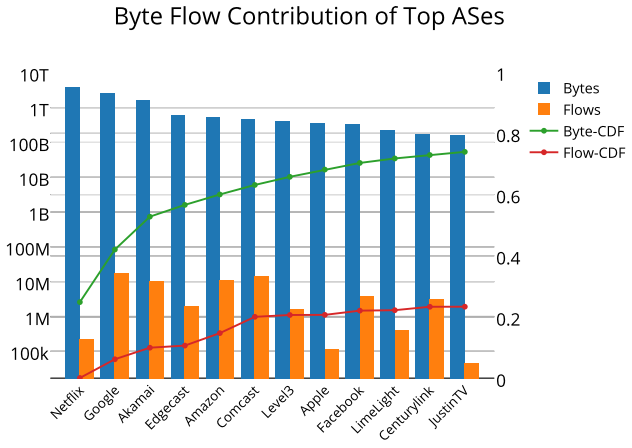

Fig. 9. Distribution of delivered Bytes and Flows towards UONet by the top 12 ASes.

AS Number	AS Name
2906	Netflix
15169	Google
20940	Akamai
15133	Edgecast
16509	Amazon
7922	Comcast
3356	Level3
6185	Apple
32934	Facebook
22822	Limelight
209	Centurylink
46489	JustinTV

TABLE V
SELECTED ASES FROM TOP ASES LIST REGARDING BOTH CRITERIA FOR THE REMAINDER OF OUR ANALYSIS.

A. Methodology

For each snapshot we extract all of the unique external IP addresses for incoming flows and map them to their corresponding AS number using the Cymru service [10]. By doing so we are able to calculate the top ASes that have the biggest contribution regarding the number of incoming bytes and flows towards UOnet. The combined list of the top 50 ASes regarding the number of incoming bytes and flows is given in Table XIII. For every given snapshot we choose a couple of well known ASes such as Google, Facebook, Akamai and Netflix which have a significant contribution in the incoming traffic of UONet and extract all of their incoming flows in order to study their content delivery mechanism. After extracting all of the incoming flows of an AS to decrease the processing overload of our measurements, we limit the granularity of our analysis to /24 prefixes. The rationale behind this decision is that /24 prefixes are most likely collocated and thereby should exhibit very similar performance metrics. For each AS, we extract all of the /24 prefixes and measure the total number of bytes and flows that were delivered from these prefixes. We choose a random valid IP address within each of these prefixes and run two sets of traceroutes towards it, one using an UDP probe and another using a TCP probe. Some ASes filter specific types of traceroutes or incoming connections and thereby we use two types of traceroutes to have a higher rate of reaching the destination server. We use the traceroute that was able to reach the destination IP address. If the traceroute was unsuccessful we choose the traceroute that was able to go further through the target ASes network. If none of the previous attempts are true for the traceroutes, we choose one of them randomly. We also map the selected IP addresses to their GEO location using Akamai's EdgeScape service. To ensure that the GEO information given using this service is accurate we compare the RTT values from our traceroutes and based on the speed of light if the GEO distance is not reachable in the given time we query the GEO information from the IP2Geo Lite database if the given value conforms to the speed of light restrictions we use this value as the GEO location for the IP address.

In summary regarding traffic locality we have the following four metrics:

1) GEO distance
2) Network Hop Distance
3) AS Distance
4) RTT

We should note that the traceroutes and IP to AS mappings are performed a week after our target daily snapshot. This would minimize any possible error caused by changing in IP to AS mappings.

Live Experiments: We also run some live experiments towards the selected ASes and pull some content from them in order to have a better understanding of the content delivery mechanisms of each AS and to extract information that might not be available from the Netflow data. Our test environment for live experiments is a Linux virtual machine. We use the Chrome browser with no extensions installed, we believe that the majority of users do not install extensions that alter the requested content from a website such as Adblock. While pulling content from each AS we collect the pcap dump of the traffic using Wireshark, we also collect the browser interaction with the website using Chrome's developer menu. The reason that we rely on both HAR files and pcap dumps is that if we

interact with a website that is delivering its content using the HTTPS protocol we will not be able to infer the content of the connection from the pcap dumps, while portions of the traffic such as content pulled from flash files are hidden from the HAR logs. The combination of these two logs allows us to have a complete picture of the interactions that the browser has conducted with our target website.

An aggregate view regarding the traffic locality of the top ASes that UONet contacts are presented in Figure 10. The radar plots presented in this Figure show the amount of locality that we observe for the selected 12 ASes according to different distance metrics which we have proposed. The ASes are sorted based on their contribution regarding the amount of Bytes towards UONet's traffic and are ordered in a clockwise manner. The 50, 75 and 90 percent values for the Bytes and Flows that have been delivered to UONet is depicted using different colors. As an example we observe that about 75% of Google's traffic is being delivered from less than 1000 Km away from UONet while to obtain the remaining 15% of traffic to reach the 90% value we have to go to a 3000 Km distance. We should also note that the distribution regarding Bytes and Flows is different since each flow does not carry the same amount of Bytes and thereby for ASes that deliver mouse flows, we can see that the 90% value is being delivered from further distances. Overall we observe a level of conformity between the GEO and RTT plots since there is a natural relationship between distance and delay this could be related to the fact that we have also checked this conformity while we were GEO locating the IP addresses. By looking at the AS hop distance plots we observe a uniform and simple story and see that the majority of UONet's traffic is being delivered from 3 AS hops away while this picture becomes more complex for the other distance metrics. This complexity is understandable given the level of granularity that each one of these distance metrics exposes. On average we observe that about 50% of traffic is being delivered from less than 1000 and 2000 Km away with respect to Bytes and Flows accordingly while except for a few cases the majority of traffic (90%) is being delivered from 10 router hops from UONet. This level of locality could somehow be explained by the GEO location of UONet with respect to the infrastructure and datacenters of the majority of content providers over today's Internet. Within the west coast these companies usually deploy their infrastructure either in Seattle, WA or Los Angeles, CA since these are close to the most populated cities of the west coast. We should note that from our test computer to UONet's border, within our traceroutes we observed an average delay of 1 ms and we have 2 router hops in between our computer and UONet's upstream providers.

B. Per AS Analysis

In this section we take a closer look at the traffic locality of each of our target ASes and present a prefix level of granularity with respect to locality, we try to see if our passive data from our Netflow dataset are in line with the live experiments that we conduct in addition to any information we could find online regarding the deployment and location of the infrastructure for each of these ASes.

After looking at the contribution of each prefix of an AS both regarding the number of bytes and flows we realized that usually only a small fraction of the prefixes are responsible for the majority of the traffic. To limit our focus towards these prefixes we only selected the outlier prefixes with respect to Bytes and Flows and also a subset of random prefixes which amounted to a minimum of 2 or 1/10 the amount of outlier prefixes as a baseline for comparison with the outlier prefixes. A prefix would be considered an outlier if its value is more than three standard deviations away from the mean value for all prefixes. After selecting the prefixes we produced a set of stacked plots which present the amount of locality we observe with respect to our distance metrics at a per prefix granularity. From top to bottom the plots present the following information:

1) Total amount of Bytes delivered from this prefix.
2) Total amount of Flows delivered from this prefix.
3) The number of IP addresses we observed within this prefix.
4) AS distance for this prefix.
5) Hop distance for this prefix.
6) GEO distance of this prefix.
7) RTT time for this prefix.

The reader should also note that since it is possible for a traceroute to fail to reach it's destination in some stack plots we could observe missing points for these prefixes. If a traceroute is able to reach a node that is residing in the destination AS we would still calculate the AS distance for that prefix.

Netflix: The general statistics regarding Netflix's incoming flows are given in Table VI.

Bytes - % of total	Flows - % of total	Prefix Count
3.8 TB - 24.84%	221.95 K - 0.08%	68

TABLE VI
GENERAL STATISTICS FOR ALL OF THE INCOMING FLOWS OF NETFLIX'S AS ON THE 2015-02-04 SNAPSHOT.

From Table XIII Netflix is the top contributor towards the total incoming bytes although it is delivering this traffic over a small number of flows. This is a key characteristic of large data flows which in Netflix's case is video content. The stacked plot for the selected prefixes of Netflix is shown in Figure 11.

By looking at the stacked plots we observe that the majority of Netflix's traffic is being delivered by prefixes number 1 and 2. These two prefixes are responsible for 94%(90%) of the traffic with respect to bytes(flows). These prefixes are located in Seattle, WA and conform to the information provided by Netflix's Open Connect database [12]. These two prefixes exhibit the lowest values regarding all of our distance metrics and thereby we could conclude that Netflix is utilizing its infrastructure to deliver content from its most local servers to UONet.

Google: The general statistics regarding Google's incoming flows are given in Table VII.

As it is evident from Table XIII Google is the top AS regarding the number of incoming flows to UONet and second regarding the number of bytes.

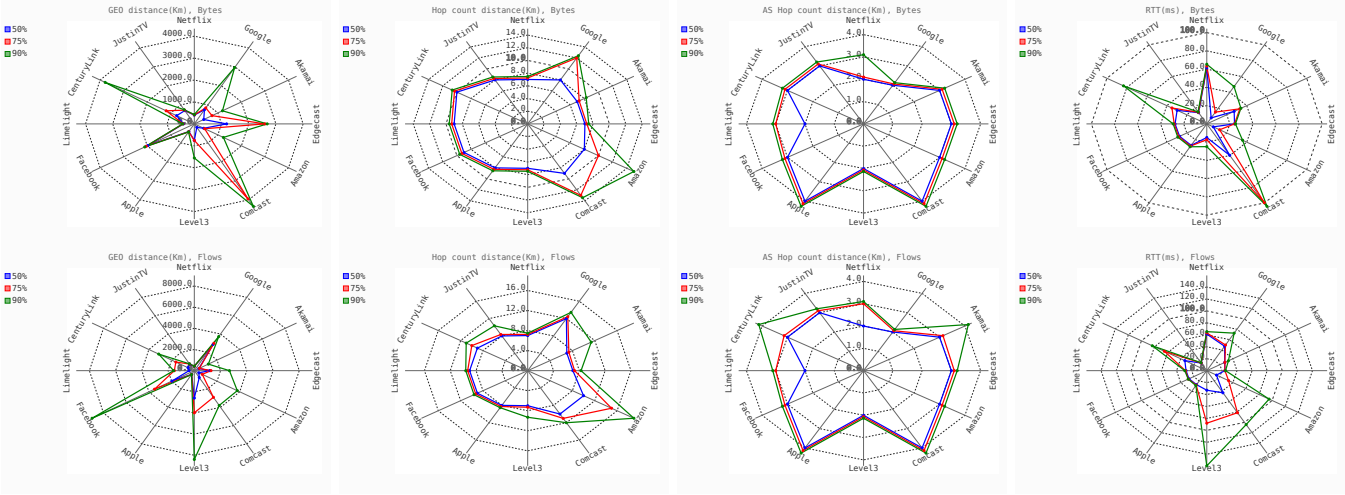

Fig. 10. Locality of traffic for top ASes, the distance that the associated fraction of traffic is delivered to UONet. From left to right representing GEO, hop count, AS hop count and RTT distance metrics accordingly.

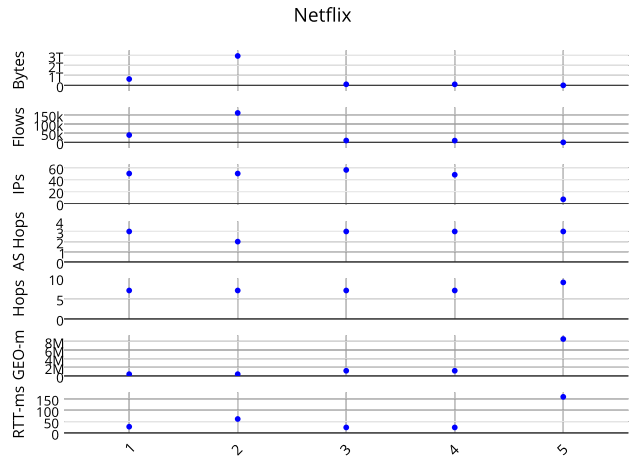

Fig. 11. stacked plot for selected prefixes of Netflix.

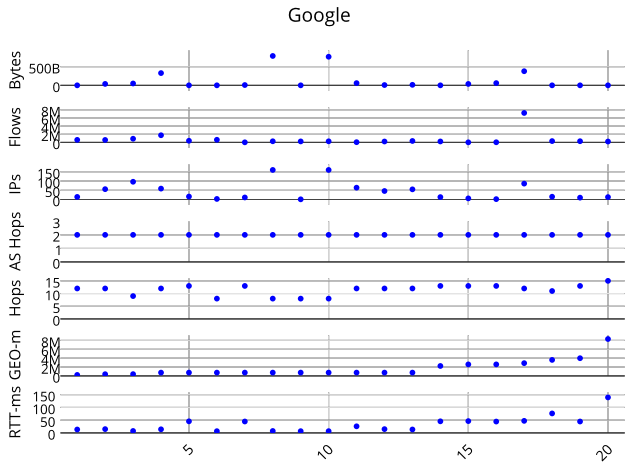

Fig. 12. stacked plot for selected prefixes of Google.

Bytes - % of total	Flows - % of total	Prefix Count
2.63 TB - 17.19%	17 M - 6.14%	1207

TABLE VII
GENERAL STATISTICS FOR ALL OF THE INCOMING FLOWS OF GOOGLE'S AS ON THE 2015-02-04 SNAPSHOT.

The stacked plot for the selected prefixes of Google is shown in Figure 12.

As it is evident from the stacked plots the majority of Google's traffic is being delivered from four prefixes (4, 8, 10 and 17) which are close to our network regarding all of our distance metrics. This finding is inline with Google's Global Cache deployment across the globe. These four prefixes are responsible for 86.12% of the incoming traffic volume from Google. In our earlier snapshots we observed that Google was delivering Youtube videos over unencrypted channels but in our latest snapshot we saw that Google has started delivering Youtube videos over encrypted channels. While running live experiments to generate traffic from Youtube we observed that the video content and the advertisements in between videos were delivered through prefixes 8 and 10. Prefix 17 was responsible for delivering files through the Google Drive service and was observed as the dominant prefix while interacting with the GMail service. Based on the stacked plots and the live experiments we could state that Google is delivering their content through local servers and their delivery mechanisms are distance aware. Although the delivery of this traffic is happening through their own AS and not through cache servers residing in other ASes. All prefixes are mapped to Mountain View, CA. Furthermore, these prefixes are located within 10-12 hops and 50ms RTT of UONet. The significant level of traffic locality for delivered content from Google is primarily due to the deployment of many Google caches across the Internet during the past few years that enables each client to receive common services and content from a close-by cache server [4]. It is worth noting that Google owns a datacenter in The Dalles, OR [13] that is at a closer GEO distance to UONet. However, we have not observed any flow from this datacenter in our NetFlow data possibly due to the services offered in that facility.

Facebook: The general statistics regarding Facebook's incom-

ing flows are given in Table VIII.

Bytes - % of total	Flows - % of total	Prefix Count
0.33 TB - 2.16%	3.82 M - 1.37%	116

TABLE VIII
GENERAL STATISTICS FOR ALL OF THE INCOMING FLOWS OF FACEBOOK'S AS ON THE 2015-02-04 SNAPSHOT.

As it is evident from Table XIII Facebook is ranked as the 9th and 11th top AS regarding the number of bytes and flows respectively.

The stacked plot for the selected prefixes of Facebook is shown in Figure 13.

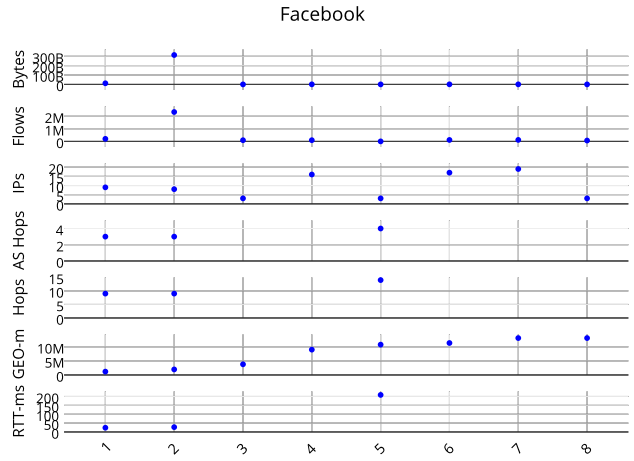

Fig. 13. stacked plot for selected prefixes of Facebook.

From the stacked plots we observe that the majority of Facebook's traffic is being delivered through one prefix, namely prefix 2. This prefix is located in Seattle WA. In our earlier snapshots we observed that Facebook was delivering most of its traffic through a prefix residing in Ireland. It seems that Facebook has improved the locality of their content delivery mechanisms. While running live experiments we observed that Facebook is delivering its video content through its own CDN while it's relying on a mixture of its own CDN and Akamai for the delivery of images that have been uploaded to Facebook. We also observed that Facebook is using Akamai's CDN to host static page elements such as Javascript files. The logic for balancing Facebook's content through their own CDN and Akamai is unknown to us and might be of interest for future research.

Akamai: The general statistics regarding Akamai's incoming flows are given in Table IX.

Bytes - % of total	Flows - % of total	Prefix Count
1.64 TB - 10.75%	10.45 M - 3.77%	7825

TABLE IX
GENERAL STATISTICS FOR ALL OF THE INCOMING FLOWS OF AKAMAI'S AS ON THE 2015-02-04 SNAPSHOT.

From Table XIII Akamai is ranked in 3rd and 5th place regarding the number of incoming bytes and flows to UONet respectively.

The stacked plot for the selected prefixes of Akamai is shown in Figure 14.

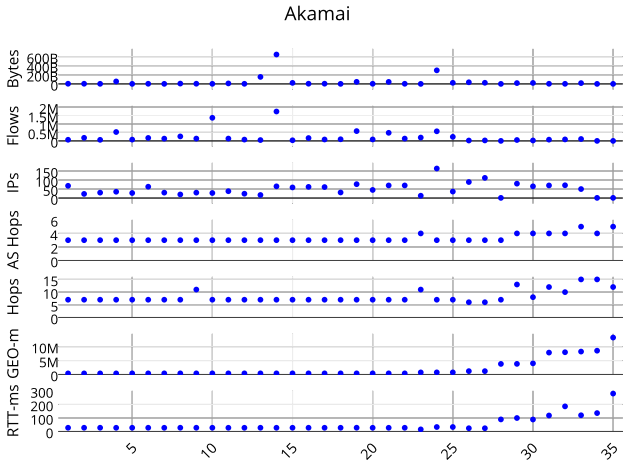

Fig. 14. stacked plot for selected prefixes of Akamai.

To pull content from Akamai's servers we selected a couple of their listed customers [14] and visited their websites to download some content from their network. From the list of customers listed on their website we selected CNN, BBC and MTV and browsed their websites and watched a couple of videos from their website to generate some traffic that is most likely delivered through Akamai's CDN. From the stacked plots we see that the majority of Akamai's traffic is being delivered through three prefixes (13, 14 and 24). These prefixes are responsible for 67% of the total volume of Akamai's traffic. While browsing CNN's website we saw that static web page elements such as images and javascript files were delivered through prefix 13 while the video content was delivered through prefix 24. For BBC we observed that static web page elements such as pictures and flash files were delivered through prefix 13, while the video content was delivered through another AS namely Centurylink. Later on in section VIII we explain how we uncover that the specified server actually belongs to Akamai while it resides in Centurylink's AS. For MTV we observed a similar scenario to BBC small page elements were delivered through prefix 13, while the video was delivered through Level3's AS. We can see that Akamai is employing a dual strategy to deliver content to UONet, at times it is employing the servers that are residing in it's own AS while it is also benefiting from servers that is has placed in other ASes.

Edgecast: The general statistics regarding Edgecast's incoming flows are given in Table X.

Bytes - % of total	Flows - % of total	Prefix Count
0.54 TB - 3.85%	1.90 M - 0.69%	144

TABLE X
GENERAL STATISTICS FOR ALL OF THE INCOMING FLOWS OF EDGECAST'S AS ON THE 2015-02-04 SNAPSHOT.

From Table XIII Edgecast is ranked in 4th and 20th place regarding the number of incoming bytes and flows to UONet respectively.

The stacked plot for the selected prefixes of Edgecast is shown in Figure 15.

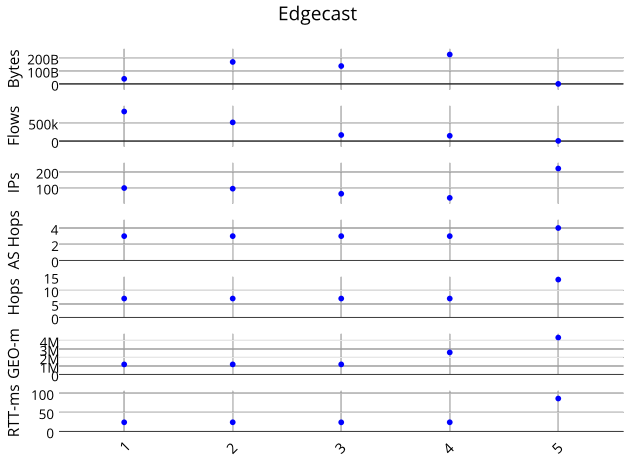

Fig. 15. stacked plot for selected prefixes of Edgecast.

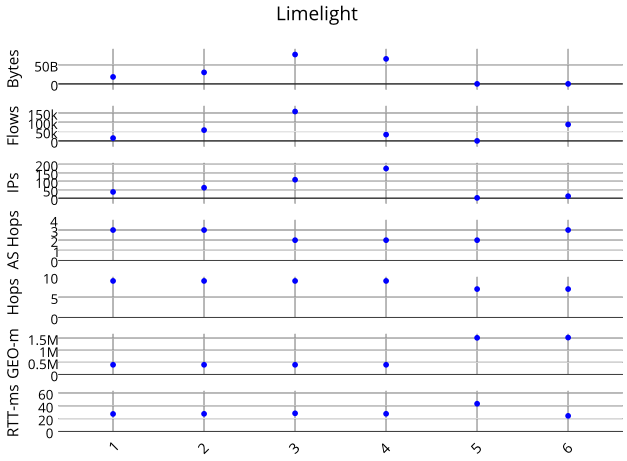

Fig. 16. stacked plot for selected prefixes of Limelight.

From the stacked plots we observe that prefixes number 2, 3 and 4 are responsible for the majority of traffic, these three prefixes are responsible for 90% of the traffic associated to Edgecast. The GEO location for prefixes 2 and 3 is San Jose, CA which is in accordance to the information that are given on Edgecast's website [15]. For prefix number 4 we observe that it is located at a GEO location which is twice as far as the two other major prefixes while its other distance metrics are very similar to prefixes 2 and 3. This prefix is GEO located to Dallas, TX while the RTT value obtained through traceroutes defies the possibility for this prefix to reside within that location. Although we have taken measures to avoid GEO mapping errors there always is a possibility for these errors to happen. Based on the other distance metrics we can assume that prefix number 4 is located in the same location as the other two prefixes and based on location of Edgecast's infrastructure [15] we can conclude that Edgecast is delivering it's content from the most local facilities with respect to UONet. For our live experiments we visited LiveLeak and browsed some videos from this website we observed that the video files along with small objects such as Javascript files were delivered through prefix 4. Through our live experiments we were not able to pull any content from the other prefixes.

Limelight: The general statistics regarding Limelight's incoming flows are given in Table XI.

Bytes - % of total	Flows - % of total	Prefix Count
0.20 TB - 1.47%	0.41 M - 0.15%	209

TABLE XI
GENERAL STATISTICS FOR ALL OF THE INCOMING FLOWS OF LIMELIGHT'S AS ON THE 2015-02-04 SNAPSHOT.

From Table XIII Limelight is ranked in 11th and 84th place regarding the number of incoming bytes and flows to UOnet respectively.

The stacked plot for the selected prefixes of Limelight is shown in Figure 16.

We can observe from the stacked plots that prefixes 2,3 and 4 are the major prefixes responsible for the delivery of Limelights traffic towards UONet. These prefixes are responsible for 77%(60%) bytes(flows) of Limelights traffic. Unfortunately we weren't able to pull that much content from Limelight's CDN network by visiting the websites of a couple of their customers we observed that most of them are relying on Akamai's CDN network. This could be a result of nonalignment between the snapshot date and the time that we were running live experiments. Among the customers that were still relying on Limelight's network all of their content regardless of its type was delivered through prefix number 2 and we did not observe any traffic from the other major prefixes. Prefix number 2 is GEO located to Seattle, WA which is inline with the information that is provided by Limelight [16]. Given the CDN locations and the distance metrics we observe from the stacked plots we can conclude that Limelight is delivering its traffic from the most local facilities.

For the purpose of brevity and since some of the top ASes are transit networks and pulling traffic from them is not as trivial as the other ASes we didn't present the remainder of the stacked plots for our target ASes. All of the stacked plots could be observed from [17].

VIII. GUEST SERVERS

In order to have a global reach and to increase the user perceived performance, websites and content providers either rely on CDN networks or deploy their own content delivery network on a scale that is economical for them and has the best reach towards its demography. The content delivery strategy could vary widely and could be as simple as hosting bulk of the traffic through a third parties CDN servers or could be as complex as managing multiple data centers on a global scale in case of Google. Netflix has recently closed all of its datacenters and is relying on an overlay network which is a mix of a central C&C center which is managed through Amazon's cloud network and delivers it's content through various cache servers that have either private peering or have been connected to other ASes at IXP's throughout the US [18]. Since it isn't possible for each CP to have a datacenter at locations which are near its customers, CP's place cache servers within various ASes to have a better reach

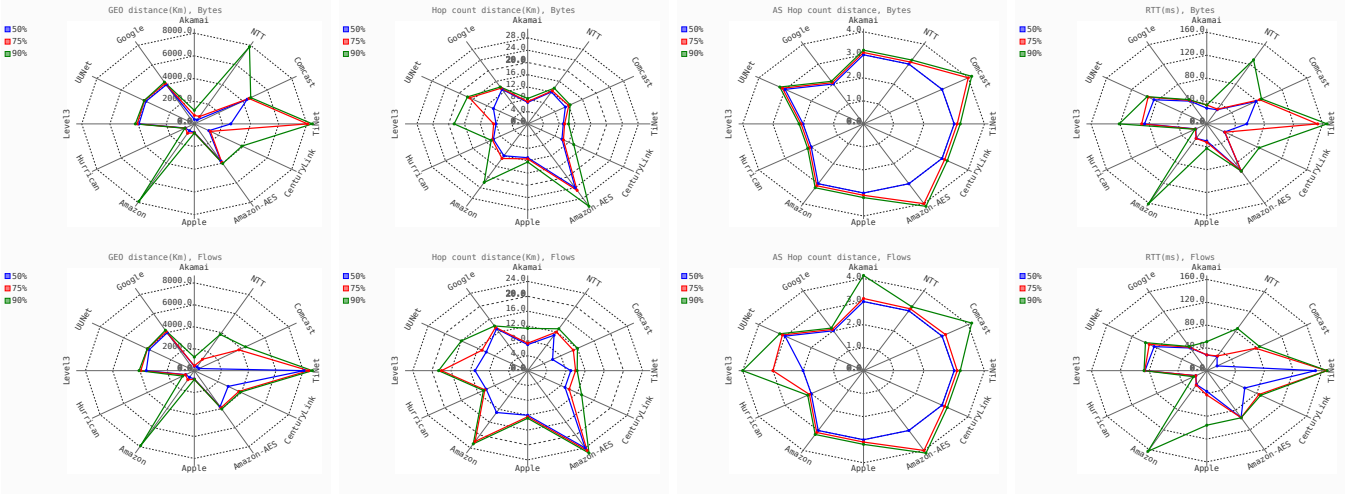

Fig. 17. Locality of traffic for Akamai's guest servers residing within the top 40 ASes in snapshot 2015-02-04.

and wider presence while avoiding the cost of maintaining big datacenters. We refer to these servers as guest servers and the AS which they reside in as the host AS. Using the Cymru service these guest servers would be mapped to their hosting AS. In this section we outline a technique which we have employed to identify the guest servers that belong to Akamai's network.

A. Methodology

Our approach for identifying Akamai servers could be broken down into two major steps:

1) Identifying Akamai Served Objects: For this step we visit some of Akamai's customers websites and identify objects within the website that are hosted by Akamai servers. We select small objects that would be accessed more frequently such as Javascript and CSS files so with a higher likelihood they would be cached through the Akamai's network.
2) Probing IP's: Through this step we would probe the IP addresses of the ASes that belong to the union of top 20 ASes regarding Bytes and Flows, and would see if the given IP address would serve our test objects. A positive response is an indicator that the given IP address is an Akamai server.

For the first step of the methodology we selected two Javascript files one from *apple.com* and another from *census.gov*. Since an Akamai server is responsible for hosting content for various websites it server should have a way to differentiate requests from multiple websites from each other. This task is achieved through the HOST field of the HTTP header. For the second step of our methodology while requesting the objects from IP addresses, we construct the HTTP request as if it was generated by a client that is visiting the host website. If a server responds with a HTTP OK/200 code we consider this IP address as an Akamai server if the request fails or timeouts we repeat the process using the other object we have selected through step 1. Through our initial tests we realized that some servers would respond with a HTTP 200 code regardless of the request that they receive. To eliminate these servers from our final results we modified our methodology and would also ask for the first 100 bytes of the object and would only consider the server as an Akamai server if the returned content matches the first 100 bytes of the original object.

To evaluate the correctness of our technique we consider all of the 22.2K servers that were mapped to Akamai by Cymru and ran our methodology over these servers. We observed that 90% of these servers were identified using our methodology. We ran a port scanner against the remainder 10% of the servers and found that only 643 (3%) of these servers had an open port to serve HTTP content and the remainder of IP addresses had different purposes. Thereby the overall accuracy of our methodology is about 97% for the test case.

Using the given methodology we probed 1.4M IP addresses which belonged to 31 ASes and were able to uncover an additional 14,121 Akamai servers that were residing in 17 different ASes that collectively delivered 552 GB through 3.23 M flows. Since Akamai's own servers were responsible for 1.5 TB of traffic the delivered content through Akamai guest servers attributes to about 25% of Akamai's traffic. Among the 17 hosting ASes 99% of the traffic of Akamai guest servers is delivered from 4 ASes namely: NTT(37.5%), Comcast(27.5%), Tinet(19.2%) and Centurylink (15%).

B. Guest Server Locality

To study the traffic locality of Akamai guest servers we produced a set of radar plots similar to the ones we presented in Section VII. As a point of reference and in order to see whether employing guest servers has increased the amount of traffic locality from UONet's standpoint of view we included a beam in these plots which corresponds to traffic locality for Akamai's own servers. Figure 17 depicts the radar plots for Akamai own and guest servers with respect to our distance metrics. Interestingly enough we observe that with respect to all of our distance metrics Akamai's own servers have a greater performance. We should note that these servers aren't necessarily employed for the purpose of traffic locality towards UONet and could be responsible for delivering content to other

stub-ASes. As we have stated, Akamai guest servers were responsible for about 25% of Akamai's total traffic which could be an indicator that we only observe traffic from these guest servers for load balancing purposes or cache misses on own servers.

IX. Performance Implications of Locality

One of the main goals of content providers for deploying a widespread infrastructure is to improve the users performance with respect to delay and bandwidth. Towards this end we study the effect of traffic locality for our target ASes. To do so we select two prefixes from each AS that have the largest contribution towards traffic with respect to bytes. Since we are relying on Netflow data for our bandwidth measurements we could not observe variations of bandwidth during the life of a flow and we can only calculate the mean value for bandwidth. Since bandwidth could also be limited by TCP congestion control algorithms specially for short lived flows we only consider flows which have more than 50 packets and have an average packet size larger than 1200 bytes. Figure 18 depicts the summary distribution for the bandwidth of the top 2 prefixes of our target ASes. The ASes are ordered according to their rank with respect to delivered bytes. We observe that on average the median value of bandwidth is relatively consistent among all of our target ASes. Interestingly Netflix which is a know provider of fat flows for video content has a lower value with respect to other ASes that deliver content of various type and sizes. One should note that the 90 percentile of bandwidth presents an upper bound for the delivery capabilities of that specific AS. As we can see there is a huge gap between the median and 90 percentile values which suggests that most of the flows did not utilize the maximum amount of bandwidth that was available at hand. This could be a side effect of many factors such as congested transit links, the type of connection on the client side or merely an indicator that the median value of bandwidth is sufficient for average requirements of a user.

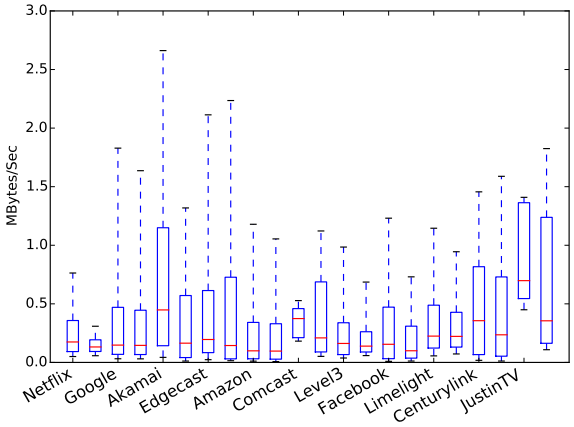

Fig. 18. summary distribution of BW for top 2 prefixes of our target ASes ordered according to AS rank regarding Bytes.

To study the effect of user connection type, we measured the summary distribution of bandwidth for the flows with more than 50 packets and average packets size larger than 1200 bytes for all of our target ASes and grouped them based on the clients connection type. Figure 19 depicts these distributions and as we can observe the median value is still consistent across different sections of the network while the 90 percentiles suggest that users which are connected through Ethernet cables were able to reach a bandwidth about twice as more than residential and wireless users. Figure 19 suggest a similar finding to Sundaresan et al. [19] that wireless links could be a bottle neck in fully utilizing the available capacity of the link.

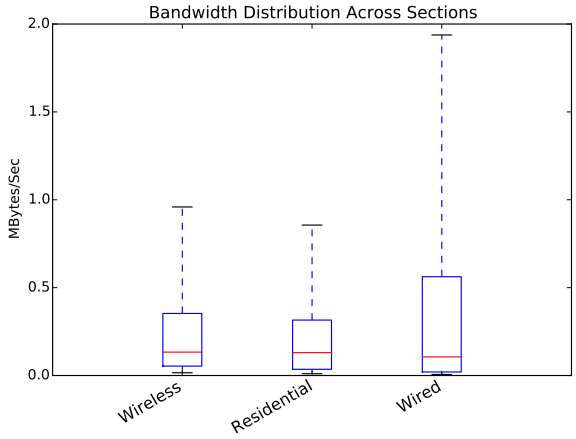

Fig. 19. summary distribution of BW for fat flows of target ASes across different sections of the network.

	RTT	GEO	HOP	AS-HOP
50% BW	-0.25 (0.27)	0.2 (0.37)	0.55 (0.01)	-0.44 (0.04)
90% BW	-0.11 (0.62)	-0.11 (0.64)	0.18 (0.43)	-0.12 (0.60)

TABLE XII
ρ (P-VALUE) FOR SPEARMAN CORRELATIONS BETWEEN 50 & 90 PERCENTILE OF FLOW BANDWIDTH FROM MAJOR PROVIDERS AND DIFFERENT MEASURES OF DISTANCE.

To study the effect of locality (GEO, hop count and RTT) on bandwidth we calculated the Spearman rank correlation between the 50 and 90 percentile values of bandwidth for the top prefixes of all of our target ASes with respect to their distance values. Table XII lists the ρ and P-values, based on these values we can conclude that there is no correlation between any of our distance metrics and the end users bandwidth. The only weak correlation is the one between the median bandwidth and hop count distance. This correlation is counter intuitive since an increase in the number of hops usually translates to higher delay values which in effect results in lower bandwidth values.

X. Related Works

Many recent studies used NetFlow data to characterize traffic originated from campus networks for network planning, performance monitoring, and troubleshooting [9], [20]–[22]. To our knowledge, none of these studies have utilized NetFlow

data to identify the top providers, their contributions into delivered traffic and their traffic footprint. Our work is also closely related to the studies that aim at characterizing the impact of geographical and network distance on the bandwidth of individual connections (*e.g.* [23], [24]). These studies focus on the performance of connections related to a single provider to infer the underlying limiting factors often using active measurement [25] or a lot more detailed information than just NetFlow data. However, our goal is to assess the impact of distance on bandwidth across flows from all providers using passive measurement. Calder et al. [4] mapped the expansion of one content provider namely Google throughout the globe and depicted their strategy to redirect users by relying on EDNS capabilities, our work does not focus on a single provider and also studies the affects of locality on bandwidth.

In a similar study [6] He et al. studied the popularity of two major cloud service providers namely Amazon EC2 and Microsoft Azure among Alexa top websites. For traffic locality the authors examined whether content providers were utilizing multiple data centers and load balancing features from cloud service providers. In our work we examine locality from a stub-AS stand point of view and look at multiple CP's.

Zink et al. [26] [27] studied the content delivery strategy of Youtube and offered suggestions for improving the performance of their infrastructure through P2P or Proxy caching. Through their work they only focus on Youtube's delivery mechanism and rely on packet headers for their analysis.

Xun et al. [22] study the dynamics of users interactions with front end servers of Google and Akamai from various vantage points by employing Planet Lab nodes and Open Resolvers. They study the latency of users to these servers and the change of mappings that happen through DNS redirections. Their work does not study the affect of users being mapped to different front end servers and their bandwidth.

Similar to our approach Sundaresan et al. [19] take a look at clients performance from the edge of the network by deploying custom monitoring software on the routers of 66 homes. Their research is mainly focused on limiting factors on the client side to fully utilize the available link capacity namely the wireless channel of the home access point. In contrast to our study they rely on packet traces instead of Netflow data. Their study does not compare and contrast the locality of traffic from the vantage point of these clients.

A recent study by Triukose et al. [25] examined whether a user is redirected to the closest Akamai servers and how the performance of various Akamai servers are different from the edge of the network using active measurement. Such an investigation clearly offers valuable insight about the strategy and performance of a single provider. Their analysis has a limited scope (*i.e.*only a single provider). More importantly, it does not incorporate the dynamics of realistic user requests from top providers and its likely interaction with the employed load balancing and cache management mechanisms by major providers. In summary, to our knowledge, this paper presents the first characterization of traffic footprint for a stub-AS and its performance implications.

XI. Conclusion

In this paper we characterized and assessed the amount of traffic locality that we would observe from a stub-AS using unsampled Netflow data from a UONet's border gateways. We presented our algorithm for defragmenting Netflow data and showed the temporal traffic trends that we have observed over the span of two years. We identified the top content providers for UONet and measured the amount of traffic locality that we observe for these major content providers. We presented traffic locality at a per AS and prefix granularity. We observed that aside for a few cases these providers exhibit a high level of locality. We presented a method for identifying *guest-servers* and using this method identified and characterized Akamai's servers residing in other ASes. Our findings indicate that Akamai's own servers offer a higher level of locality from UONet's stand point of view. At the end we assessed the affect of locality on UONet clients bandwidth and measured the bandwidth of users for the fat flows of top content providers, we did not observe any strong correlation between our distance metrics and users bandwidth for the major providers. This indicates that a combination of other subtle factors have an effect on users bandwidth.

In the future we would like to explore the following tasks: We would like to further investigate the limiting causes of users bandwidth, namely we would like to study the effect of Internet routes by studying users bandwidth over the span of time and periodically conduct traceroutes towards our target ASes. We would also like to explore strategies to associate Akamai flows to Akamai's customer using temporal patterns between UONet users and CP servers.

ASN	AS Name	Byte Rank	Flow Rank	Bytes	Flows
2906	NETFLIX	1	157	3.46 TB	220.95 K
15169	GOOGLE	2	1	2.39 TB	16.94 M
20940	AKAMAI-ASN1	3	5	1.50 TB	10.45 M
15133	EDGECAST	4	20	548.15 GB	1.90 M
16509	AMAZON-02	5	4	485.25 GB	11.29 M
7922	COMCAST-7922	6	2	442.66 GB	14.74 M
3356	LEVEL3	7	22	378.44 GB	1.66 M
6185	APPLE-AUSTIN	8	276	337.35 GB	112.33 K
32934	FACEBOOK	9	11	308.43 GB	3.82 M
2914	NTT-COMMUNICATIONS-2914	10	24	262.64 GB	1.51 M
22822	LLNW	11	84	209.12 GB	413.54 K
209	CENTURYLINK-US-LEGACY-QWEST	12	13	160.44 GB	3.10 M
46489	JUSTINTV	13	557	146.62 GB	46.22 K
20446	HIGHWINDS3	14	83	145.24 GB	422.81 K
25	UCB	15	366	123.19 GB	78.48 K
14618	AMAZON-AES	16	7	117.75 GB	7.32 M
3257	TINET-BACKBONE	17	44	114.33 GB	832.64 K
10343	NASA-AERONET-AS	18	20677	112.10 GB	46.00
40428	PANDORA-EQX-SJL	19	54	108.77 GB	693.76 K
1273	CW	20	65	102.70 GB	606.20 K
4436	AS-GTT-4436	21	34	98.01 GB	1.08 M
54888	TWITTER-NETWORK	22	46	90.46 GB	796.99 K
54113	FASTLY	23	35	86.16 GB	1.08 M
160	U-CHICAGO-AS	24	360	63.50 GB	79.69 K
12989	HWNG	25	207	61.99 GB	161.70 K
8075	MICROSOFT-CORP-MSN-AS-BLOCK	26	8	56.48 GB	6.28 M
32590	VALVE-CORPORATION	27	90	56.40 GB	387.09 K
2902	WN-WY-AS	28	2601	54.85 GB	4.41 K
13335	CLOUDFLARENET	29	30	49.47 GB	1.24 M
26769	BANDCON	30	118	47.98 GB	284.97 K
54994	WANGSU-US	31	317	47.25 GB	96.33 K
4134	CHINANET-BACKBONE	32	3	46.62 GB	13.98 M
6453	AS6453	33	120	44.38 GB	279.29 K
4837	CHINA169-BACKBONE	34	10	44.02 GB	4.58 M
36408	CDNETWORKSUS-02	35	59	42.94 GB	668.08 K
11537	ABILENE	36	563	41.58 GB	45.54 K
2828	XO-AS15	37	82	39.47 GB	423.06 K
36351	SOFTLAYER	38	16	33.20 GB	2.17 M
714	APPLE-ENGINEERING	39	6	32.55 GB	8.74 M
31976	REDHAT-0	40	2025	30.49 GB	6.72 K
5511	OPENTRANSIT	41	199	29.99 GB	168.48 K
6461	ABOVENET	42	42	29.59 GB	864.95 K
21859	C3	43	479	28.17 GB	55.99 K
6762	SEABONE-NET	44	191	27.54 GB	178.50 K
3701	NERONET	45	161	24.77 GB	216.14 K
174	COGENT-174	46	55	22.68 GB	693.74 K
3491	BTN-ASN	47	248	21.30 GB	129.74 K
7843	TWCABLE-BACKBONE	48	185	21.16 GB	183.31 K
23456	failed AS res	49	14	20.95 GB	2.58 M
20001	ROADRUNNER-WEST	50	98	19.83 GB	353.18 K
23650	CHINANET-JS-AS-AP	107	9	6.19 GB	5.29 M
21342	AKAMAI-ASN2	464	12	444.38 MB	3.47 M
6939	HURRICANE	54	15	18.11 GB	2.45 M
0	failed AS res	605	17	273.17 MB	2.09 M
33517	DYNDNS	499	18	394.02 MB	2.03 M
701	UUNET	68	19	11.35 GB	1.91 M
7029	WINDSTREAM	211	21	1.65 GB	1.66 M
20115	CHARTER-NET-HKY-NC	55	23	17.84 GB	1.52 M
7018	ATT-INTERNET4	70	25	11.13 GB	1.49 M
3462	HINET	94	26	7.38 GB	1.38 M
4808	CHINA169-BJ	136	27	3.52 GB	1.34 M
23724	CHINANET-IDC-BJ-AP	116	28	4.59 GB	1.30 M
10439	CARINET	867	29	138.60 MB	1.26 M
12876	AS12876	110	31	5.44 GB	1.22 M
8560	ONEANDONE-AS	191	32	2.00 GB	1.18 M
36647	YAHOO-GQ1	59	33	14.89 GB	1.16 M
8151	Uninet	327	36	801.24 MB	1.06 M
12182	INTERNAP-2BLK	105	37	6.20 GB	997.91 K
22561	CENTURYLINK-LEGACY-LIGHTCORE	485	38	411.18 MB	967.92 K
16276	OVH	57	39	16.85 GB	914.84 K
6423	EASYSTREET-ONLINE	643	40	241.74 MB	887.39 K
29990	ASN-APPNEXUS	142	41	3.32 GB	866.87 K
36089	OPENX-AS1	215	43	1.60 GB	846.98 K
8972	PLUSSERVER-AS	231	45	1.40 GB	815.46 K
13414	TWITTER	67	47	11.51 GB	793.01 K
27281	QUANTCAST	397	48	583.94 MB	788.66 K
19679	DROPBOX	85	49	8.39 GB	762.96 K
6327	SHAW	98	50	6.55 GB	741.68 K

TABLE XIII

TOP ASES WITH THEIR CORRESPONDING STATISTICS FOR THE 2015-02-04 SNAPSHOT

REFERENCES

[1] I. Castro, J. C. Cardona, S. Gorinsky, and P. Francois, "Remote peering: More peering without internet flattening," in *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*. ACM, 2014, pp. 185–198.

[2] R. V. Oliveira, B. Zhang, and L. Zhang, "Observing the evolution of internet as topology," in *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4. ACM, 2007, pp. 313–324.

[3] A. Singhal and M. Cutts, "Official Google webmaster central blog: Using site speed in Web search ranking," http://googlewebmastercentral.blogspot.com/2010/04/, [Online; accessed 30-July-2015].

[4] M. Calder, X. Fan, Z. Hu, E. Katz-Bassett, J. Heidemann, and R. Govindan, "Mapping the expansion of google's serving infrastructure," in *Proceedings of the 2013 conference on Internet measurement conference*. ACM, 2013, pp. 313–326.

[5] "Akamai Technologies Facts & Figures," https://www.akamai.com/us/en/about/facts-figures.jsp, [Online; accessed 10-July-2015].

[6] K. He, A. Fisher, L. Wang, A. Gember, A. Akella, and T. Ristenpart, "Next stop, the cloud: Understanding modern web service deployment in ec2 and azure," in *Proceedings of the 2013 conference on Internet measurement conference*. ACM, 2013, pp. 177–190.

[7] "Ubuntu Manpage: nfcapd - netflow capture daemon," http://manpages.ubuntu.com/manpages/intrepid/man1/nfcapd.1.html, 2014, [Online; accessed 31-July-2014].

[8] "library for reading netflow records from nfdump files," https://github.com/switch-ch/nfdump-libnfread, 2014, [Online; accessed 29-September-2014].

[9] E. Glatz and X. Dimitropoulos, "Classifying internet one-way traffic," in *Proceedings of the 2012 ACM conference on Internet measurement conference*. ACM, 2012, pp. 37–50.

[10] T. Cymru, "Ip to asn mapping," *h ttp://www. team-cymru. org/Services/ip-to-asn. html*, 2008.

[11] "Top Sites in the United States," http://www.alexa.com/topsites/countries/US, 2014, [Online; accessed 17-March-2014].

[12] "Netflix Open Connect Content Delivery Network," https://openconnect.netflix.com/peeringLocations/, 2015, [Online; accessed 20-August-2015].

[13] "Google Data center locations," http://www.google.com/about/datacenters/inside/locations/index.html, [Online; accessed 30-July-2015].

[14] "Customer List — Akamai," http://www.akamai.com/html/customers/customer_list.html, 2015, [Online; accessed 07-April-2015].

[15] "CDN Locations — EdgeCast," http://www.edgecast.com/network/map/, [Online; accessed 30-July-2015].

[16] "Locations — Limelight Networks," http://www.limelight.com/company/locations/, [Online; accessed 30-July-2015].

[17] "Stacked Plots of Target ASes," https://ix.cs.uoregon.edu/~byeganeh/tau/prefix-locality/, 2015, [Online; accessed 30-January-2015].

[18] "Netflix to Pull Plug on Final Data Center - The CIO Report - WSJ," http://blogs.wsj.com/cio/2015/08/13/netflix-to-pull-plug-on-final-data-center/, [Online; accessed 30-July-2015].

[19] S. Sundaresan, N. Feamster, and R. Teixeira, "Measuring the performance of user traffic in home wireless networks," in *PAM*. Springer, 2015, pp. 305–317.

[20] E. Glatz, S. Mavromatidis, B. Ager, and X. Dimitropoulos, "Visualizing big network traffic data using frequent pattern mining and hypergraphs," *Computing*, vol. 96, no. 1, pp. 27–38, 2014.

[21] D. Schatzmann, W. Mühlbauer, T. Spyropoulos, and X. Dimitropoulos, "Digging into https: flow-based classification of webmail traffic," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 2010, pp. 322–327.

[22] K. Xu, Z.-L. Zhang, and S. Bhattacharyya, "Profiling internet backbone traffic: behavior models and applications," in *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4. ACM, 2005, pp. 169–180.

[23] A. D'Alconzo, P. Casas, P. Fiadino, A. Bar, and A. Finamore, "Who to blame when youtube is not working? detecting anomalies in cdn-provisioned services," in *IWCMC*. IEEE, 2014, pp. 435–440.

[24] X. Fan, E. Katz-Bassett, and J. Heidemann, "Assessing affinity between users and cdn sites," in *Traffic Monitoring and Analysis*. Springer, 2015, pp. 95–110.

[25] S. Triukose, Z. Wen, and M. Rabinovich, "Measuring a commercial content delivery network," in *WWW*. ACM, 2011, pp. 467–476.

[26] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Characteristics of youtube network traffic at a campus network–measurements, models, and implications," *Computer Networks*, vol. 53, no. 4, pp. 501–514, 2009.

[27] ——, "Watch global, cache local: Youtube network traffic at a campus network: measurements and implications," in *Electronic Imaging 2008*. International Society for Optics and Photonics, 2008, pp. 681 805–681 805.