A Hybrid Network Traffic Engineering System

Zhenzhen Yan*, Chris Tracy†, and Malathi Veeraraghavan*
* Dept. of Electrical and Computer Eng., University of Virginia
Charlottesville, VA 22904–4743
Email: {zy4d,mvee}@virginia.edu
† Energy Sciences Network (ESnet), Lawrence Berkeley National Laboratory
Berkeley, CA 94720
Email: ctracy@es.net

***Abstract*—This paper describes traffic analysis undertaken to answer certain questions needed to design a hybrid network traffic engineering system (HNTES). The hybrid network in question consists of an IP-routed network and a dynamic virtual-circuit network, and the role of HNTES is to identify and redirect α-flows, which are defined as flows in which the number of bytes exceeds a threshold H (1 GB) over at least one α-interval (1 minute). NetFlow data from ESnet was analyzed. Our findings show that raw IP α-flows (identified by the 5-tuple) are mostly short-lived (80% are shorter than 2 minutes), which implies that HNTES should use an offline mechanism for identifying α-flows and preconfiguring policy-based routes to redirect packets for these flows to virtual circuits since setup delay is about 1 minute. Prefix flows, which are aggregates of raw IP flows, identified by /24 subnet IDs, show persistency with 16% of these flows appearing in more than quarter of the days in the observed period.**

I. Introduction

Scientific computing applications used in fields such as high-energy physics, climate science, genomics, etc., generate large (tera- to peta-byte sized) data sets. These data sets need to be moved from the supercomputing facilities on which the computing applications are executed to servers located at the various scientists' universities and laboratories.

To offer scientists a rate-guaranteed high-speed file transfer service, core research-and-education networks (RENs), such as the Department of Energy (DOE)'s Energy Sciences Network (ESnet) [1] and Internet2 [2], are now offering a new type of connectivity service, i.e., a dynamic circuit service[1], as a complement to their IP-routed service. ESnet operates separate networks for these two services, using the name "Science Data Network (SDN)" [3] for its virtual-circuit network. Such dynamic circuit services are also offered by commercial providers, such as AT&T and Verizon [4].

While core networks, such as ESnet and Internet2, support the dynamic virtual circuit service, regional RENs and campus networks lag behind. To realize the benefits of using this service for large file transfers, i.e., low throughput variability, the virtual circuits need to extend end-to-end. Therefore, the usage of these core dynamic circuit networks and service has been limited. This has led to an identification of a different usage model, one that has value even when the virtual circuits are just intra-domain in scope.

This alternative usage of a core virtual circuit service is as follows. Core providers have recognized that some scientific users have the required end equipment (cluster computers with parallel file systems and high-speed disk arrays), as well as high-speed links end-to-end on their IP-routed paths, to cause their TCP senders to increase their sending rates to significant fractions of the core network link capacities. For example, with high-end computing and storage devices, scientists can move files end-to-end at multiple, e.g, 2-3, Gbps, which is a significant fraction of the typical core network link capacity, which is 10 Gbps. Such transfers cause spikes in the loading conditions of the core IP-routed network links, which in turn, have adverse effects on general-purpose flows carried on the links.

To handle this problem, core providers are interested in developing and deploying hybrid network traffic engineering systems (HNTES) that can (i) identify these high-rate flows as they enter the core network at ingress IP routers, and (ii) redirect them to their deployed, but lightly used, virtual-circuit networks. In other words, by offloading these high-rate flows from the IP-routed network to the virtual-circuit network, core providers expect to have fewer reports of problems from their general-purpose users. Solutions to this problem only require intra-domain deployment, and can hence yield the immediate benefits of reduced operational costs.

Section II provides background information presenting the specific problem addressed in this work. Section III reviews related work. Alternatives for several aspects of a HNTES design are discussed in Section IV. To choose between these alternatives, traffic analysis is required. Section V describes our findings from an analysis of ESnet traffic data. The paper is concluded in Section VI.

II. Background and problem statement

Background information on the terminology used, NetFlow, policy based routes, virtual circuit setup delay, and flow characteristics, is provided before presenting the problem statement of this work.

Terminology. A flow is identified by the 5-tuple {source IP address, destination IP address, source port number, destination port number, protocol type}. Such a flow is referred to

[1]Different providers use circuit (such as SONET/WDM) or virtual circuit (such as MPLS) networks to support this service. In this paper, the terms "circuit" and "virtual circuit" are used interchangeably.

as a *raw IP flow* in [5]. A second term, *prefix flow*, is also introduced in [5] to characterize aggregates of raw IP flows. For example, a prefix flow can be defined as the aggregate of all raw IP flows between the same source and destination IP addresses. Another example is a prefix flow that aggregates raw IP flows from hosts in a source subnet to hosts in a destination subnet. This terminology of "raw IP flows" and "prefix flows" is adopted in this work.

NetFlow. Part of the task of classifying packets into flows is done by NetFlow [6], which is a feature available in most IP routers. NetFlow enables IP routers to collect a sample of packet headers, which carry the 5-tuple flow identification information described above. Each IP router's NetFlow system maintains a running set of flow reports. For each such report, it maintains the timestamp of the first and last packets as well as the total flow size. At the end of each *active timeout interval*, which is typically set to 60 seconds, the stored flow reports are exported from the IP router to a *collector*. Processing and maintaining volumes of NetFlow data can be both computationally and storage intensive, especially for routers with high-speed links. Therefore, packet sampling is used, e.g., NetFlow is configured to sample 1-in-100 and 1-in-1000 packets in Internet2 routers and ESnet routers, respectively. The associated drawback lies in the accuracy of flow feature estimates made from NetFlow data. Nevertheless, the starting point for our flow identification algorithm is NetFlow data.

Policy-Based Routing (PBR). Another feature of routers that will be leveraged in our system design is support for flow redirection. The PBR feature allows administrators to configure alternate routes other than the typical IP route for packets belonging to specific flows, which could be raw IP flows or prefix flows. The PBR table is consulted before the IP routing table to determine how to forward incoming packets. This PBR feature can be used in our application in the following manner. After determining the identifiers of heavy-hitter flows, the traffic engineering system can set PBR entries in the IP routers to cause packets from these flows to be redirected automatically to virtual circuits. For operational reasons, e.g., troubleshooting in case of performance problems or failures, administrators would prefer to limit the number of policy based routes.

Virtual circuit setup overhead. In current deployments, such as ESnet's virtual circuit network, virtual circuit setup delays can be as large as 1 minute. The type of flows that cause adverse effects on general-purpose flows are those whose senders are capable of sending data at a rate that is a significant fraction of link capacity. Such flows may not be of long durations, which could make the 1 minute virtual circuit setup delay overhead significant. For example, a TCP sender of a 10 GB file may be able to send the data at say 2 Gbps (with high-end computing and storage at the end hosts), but such a transfer would only last a few seconds. The implication is that online (dynamic) virtual circuit setup (after packets of these high-rate flows are identified at ingress IP routers in real-time) may not be effective because these flows could end before the virtual circuit is set up. The implications of this constraint on our traffic engineering system are considered in Section IV.

Flow characteristics. For classification purposes, four dimensions of flows have been identified in [7]: size, duration, rate, and burstiness. For purposes of our intended application, i.e., hybrid network traffic engineering, a dimension identified in [8] is more appropriate. Flows are classified as "alpha-flows" if the number of bytes exceeds a threshold within a prespecified (small) duration of its lifetime. Such flows are shown in [8] to be the primary source of burstiness of IP traffic, and further the cause of such flows is identified to be large file transfers across high-speed bottleneck link paths. The latter match exactly the type of large scientific data transfers seen on ESnet. These have been identified by ESnet administrators as the primary cause of performance degradation for general-purpose flows, and are hence the primary targets for flow redirection to virtual circuits. Therefore, alpha flows (for which we use the notation α-flows) will be the ones selected for redirection.

Problem statement. There are many aspects in the design of a hybrid network traffic engineering system that can identify and redirect α-flows from the IP-routed network to a virtual circuit network. Design alternatives for some of these aspects are considered in Section IV. Based on this discussion, a set of hypotheses about the core network traffic is formulated. Section V describes our approach for testing these hypotheses, and our findings from ESnet traffic data.

III. Related work

Several papers report on elephant flows, such as [5], [7]–[9], though the definition of elephant flows varies. In [7], elephant flows are defined as flows whose size (number of bytes transferred) is larger than some threshold. But [5] defines elephant flows to be the top-ranked flows that send the most number of bytes within 1-minute intervals. This is consistent with the definition on α-flows in [8], which is a concept adopted in our work. The difference between our work and that of [5] is primarily in time scale. The persistency of raw IP flows that are elephants is noted to be high within the elephant flows' lifetimes in [5]. The authors note "once an elephant" implies "always an elephant." Conclusions about persistency on a longer time scale and for prefix flows are less clear in [5]. The definition of elephant flows in [9] is more complex in that the threshold is changed at every time interval to meet a requirement that the aggregate elephant traffic is 80% of the total traffic load.

More generally, flow classification algorithms have been of interest for reasons such as identifying the type of applications that generate the most traffic [10]. A survey paper [11] groups flow classification methods into three categories: (i) Port based, (ii) Payload based, and (iii) Flow statistics based. Our solution falls in the last category, that of using statistical properties of flows. While our flow identification technique is designed specifically for our intended application, i.e., hybrid network traffic engineering, algorithms designed for more

open-ended applications, typically use machine learning techniques because of the "large datasets and multi-dimensional spaces of flow and packet attributes" [11].

IV. Traffic engineering system design considerations

Listed below are a few questions that need answers before designing HNTES.

1) Should packets entering a router be mirrored to a HNTES server for flow identification, or should NetFlow data, in spite of packet sampling, be used?
2) Should raw IP flows or prefix flows be redirected?
3) Should α-flow identification be online or offline, i.e. should identification algorithms be executed after or before packets appear at an ingress router? If flow identification is online, then correspondingly, circuit setup and PBR configuration will need to be online.
4) If flow identification is offline, it implies circuits should be set up a priori. If so, how should link capacity from an ingress router to its core network neighbors be divided among $(N-1)$ virtual circuits, where N is the number of routers in the core network?

Question 1: NetFlow data usability. For the first question, mirroring packets to a server and performing header extraction and flow identification externally appears infeasible at high rates. To answer whether NetFlow data is sufficient to identify α-flows in spite of it being sampled data, an experimental study was conducted.

Two high-end hosts, `anl-diskpt1` located at Argonne National Laboratory, Chicago, and `lbl-diskpt1` located at Lawrence Berkeley Laboratory, Berkeley, were used to run a GridFTP server and GridFTP client, respectively. There are eight ESnet IP routers on the path between these hosts, with 10 Gb/s links between the routers. This experimental setup is such that high rates of data transfer can be sustained as the disks in the end hosts have multi-Gbps access rates and the bottleneck link rate on the end-to-end path is 10 Gbps.

GridFTP transfers of known sizes were executed between the server and client, and NetFlow data was collected from two transit routers. From the GridFTP logs stored at the server, the TCP ports of the data connections were obtained. All flow records corresponding to the GridFTP transfers were filtered out using the five-tuple identifier, and the size of each transfer was estimated using the 1000 factor multiplier on the total size reported by the flow records for the data connection, since the ESnet NetFlow sampling rate is 1-in-1000. A `size-accuracy ratio` is defined to be the ratio of the NetFlow estimated size and actual file size. For each file size (100MB, 1GB, and 10 GB), multiple runs were executed since the packet sampling at the router makes the size-accuracy ratio a random variable. For all three file sizes, the sample mean shows a size-accuracy ratio close to 1, and more interestingly, the standard deviation is smaller for larger files (0.28 for 100 MB files to 0.04 for 10 GB files).

These findings show that NetFlow data can not only be used to detect flows of large size, but can also be used for the estimation of sizes. Since NetFlow data is based on random packet sampling, and large-sized flows across high-rate paths have more packets within the NetFlow active timeout intervals, the probability of packets from these flows being captured is higher than those from small-sized flows or large-sized flows across low-rate paths.

Questions 2 and 3. The second and third questions at the start of this section are discussed together. Raw IP flow identifiers include transport-layer port numbers, which are ephemeral for most file transfer applications. If raw IP flow identifiers are used to set PBRs, flow identification needs to online. A design consisting of online flow identification, and correspondingly dynamic circuit setup and PBR configuration, is feasible if the flow durations are considerably longer than circuit setup delay.

If this is not the case, flow identification, circuit provisioning, and PBR configuration, should be performed offline. In such a design the selected flow identifiers should stay constant over long periods of time, and there should be some level of persistency in the arrival of these flows. Given the first criterion, prefix flow identifiers are better suited than raw IP flows since the latter include the ephemeral port numbers. The next question is whether prefix flow identifiers should be /32 source and destination IPv4 addresses, or source and destination subnet identifiers, e.g, /24, addresses?

Questions 4. If circuits are provisioned a priori, and such circuits are required between all ingress-egress router pairs, will the division of the link capacity between $(N-1)$ virtual circuits, where N is the number of ingress/egress routers in the core network, lower file transfer throughput? If the virtual circuit network is an MultiProtocol Label Switching (MPLS) network, rateless virtual circuits can be used [12]. By not rate limiting Label Switched Paths (LSPs), which is the name used for virtual circuits in an MPLS network, $(N-1)$ LSPs can be provisioned from each ingress router to each egress router, and yet any α-flow can burst to the full link capacity if it is the only flow occurring at that time.

Hypotheses. Toward answering the questions raised above, we formulate a set of hypotheses for testing with traffic data. These are as follows:

1) Durations of raw IP α-flows are not several multiples of circuit setup delay.
2) The number of prefix flows that contain α-flows for which PBRs are configured is not that large, making it manageable from an operational point of view.
3) Prefix flows do show some level of persistency.

V. Traffic analysis

A. Traffic analysis approach

First, a few terms are defined. An *α-flow* is defined as a raw IP flow in which the number of bytes exceeds a threshold H over at least one *α-interval*, denoted t_1, within its lifetime.

The number of bytes sent in that interval is referred to as α-*bytes*[2]. Ideally, the α-interval should be as small as possible. Given that the algorithm will use NetFlow data, t_1 is set to equal the NetFlow active timeout interval. In addition to the α-*interval*, t_1, two other time intervals are used in this algorithm: (i) t_2 is the time period over which flow reports are aggregated into raw IP flows, and raw IP flows are aggregated into prefix flows, and is hence referred to as the *aggregation interval*, and (iii) t_3 is a *monitoring interval*. Parameter τ denotes the number of *aggregation intervals* in the *monitoring interval*, i.e., $\tau = (t_3/t_2)$.

Three sets are defined: $\mathbf{R}_i$, the set of NetFlow reports for all α-intervals, $\mathbf{F}_i$, the set of unique raw IP flows, and $\mathbf{P}_i$, the set of unique prefix flows, where $1 \leq i \leq \tau$. For example, if the *AI* is one day, τ is the number of days in the monitoring interval. Table I shows the notation used.

- NetFlow reports: By definition, the α-bytes of all NetFlow reports in sets $\mathbf{R}_i$ are lower-bounded by H, i.e.,

$$\beta_{ij} \geq H,\ 1 \leq i \leq \tau,\ 1 \leq j \leq m_i \tag{1}$$

- Raw IP flows: Each raw IP flow $\mathbf{f}_{ik}$, $1 \leq k \leq n_i$, as listed in Table I, is created by aggregating a subset of reports from $\mathbf{R}_i$. The common feature of all the reports in this subset is that they share a common identifier, that of the raw IP flow γ_{ik}. Thus, a set of indices $\mathbf{J}_{ik} = \{j_1, j_2, \cdots, j_{c_{ik}}\}$ is selected from report set $\mathbf{R}_i$ such that the identifiers $\omega_{ij_a} = \gamma_{ik}$ for $1 \leq a \leq c_{ik}$ and $j_a \in (1, m_i)$. For each raw IP flow, $\mathbf{f}_{ik}$ the α-bytes

$$\delta_{ik} = \sum_{j=1}^{c_{ik}} \beta_{ij}, \text{s.t.}, j \in \mathbf{J}_{ik},\ 1 \leq i \leq \tau,\ 1 \leq k \leq n_i \tag{2}$$

  The α-time of a raw IP flow is

$$\xi_{ik} = \sum_{j=1}^{c_{ik}} (e_{ij} - s_{ij}), \text{s.t.}, j \in \mathbf{J}_{ik},\ 1 \leq i \leq \tau,\ 1 \leq k \leq n_i \tag{3}$$

- Prefix flows: First, a procedure is required to define a set of prefix flows $\mathbf{P}_i$ by aggregating raw IP flows from the set $\mathbf{F}_i$, for each *aggregation interval (AI)* i, where $i \in (1, \tau)$. Assume that the indices of the raw IP flows in the set $\mathbf{F}_i$ aggregated into prefix flow $\mathbf{p}_{il}$ is denoted $\mathbf{K}_{il} = \{k_1, k_2, \cdots, k_{g_{il}}\}$, $1 \leq l \leq d_i$. The identifiers γ_{ik_a} of the corresponding raw IP flows should all be aggregatable into the prefix flow identifier ζ_{il}, for $1 \leq a \leq g_{il}$ and $1 \leq k_a \leq n_i$. Moreover, the ingress router IDs, $\Re^I_{ik_a}$, of these raw IP flows are all the same and equal to $\Re^I_{il}$, and the egress router IDs, $\Re^E_{ik_a}$, of these raw IP flows are all the same and equal to $\Re^E_{il}$. For each prefix flow, the α-*bytes* value is determined as follows:

$$\eta_{il} = \sum_{k=1}^{g_{il}} \delta_{ik}, \text{s.t.}, k \in \mathbf{K}_{il},\ 1 \leq i \leq \tau,\ 1 \leq l \leq d_i \tag{4}$$

[2]The term "size" is not used to avoid confusion with the term "size of a flow," which is the aggregate size, in bytes, of all packets within a flow's duration.

The next step is to determine the α-*time* of each prefix flow (see Table I). While the α-intervals of a raw IP flow are necessarily non-overlapping (given how NetFlow formulates its reports), α-intervals from different raw IP flows that are part of the same prefix flow can have overlaps. To find the α-time of a prefix flow, the overlapping intervals should be merged to find the total time across an AI in which a prefix flow has a constituent raw IP flow experiencing an α-interval. The α-intervals of each prefix flow $\mathbf{p}_{il}$ are divided into two sets: $\mathbf{O}_{il}$ consisting of x_{il} overlapping α-intervals, and $\mathbf{N}_{il}$ consisting of y_{il} non-overlapping α-intervals. A new set of non-overlapping intervals $\mathbf{M}_{il}$ of size u_{il} is derived from $\mathbf{O}_{il}$ as follows: from a contiguous set of overlapping α-intervals within set $\mathbf{O}_{il}$, a new interval is created for set $\mathbf{M}_{il}$ with the earliest start time, s^e_{iv}, and the latest end time, e^l_{iv}, $v \in (1, u_{il})$. The α *time* is then computed as

$$\mu_{il} \triangleq \sum_{v=1}^{u_{il}} (e^l_{iv} - s^e_{iv}) + \sum_{u=1}^{y_{il}} (e_{iu} - s_{iu}),$$
$$1 \leq i \leq \tau,\ 1 \leq l \leq d_i \tag{5}$$

B. Traffic analysis findings

ESnet NetFlow data (which uses 1-in-1000 packet sampling) was collected from one ESnet provider edge router. Data was collected for two months: July and August 2011 (62 days). NetFlow samples were limited to flows entering ESnet at this router. Flow tools [13], Perl and R [14] programs were used to analyze the data.

The following parameter values were used (see above subsection for the meaning of these parameters): t_1 is 1 minute; t_2 is 1 day; t_3 is 62 days; and H is 1 GB. Two types of prefix flows are used: /32 source and destination IP addresses, and /24 source and destination subnets.

The total number of raw IP flows, total α-bytes, and total α-time, is plotted on a per-day basis in Fig. 1. All three values peaked on day 28 and 29, e.g., on day 28, there were 659 raw IP flows, 2.65 TB α-bytes, and 33416 seconds of α-time.

Fig. 2 shows a histogram of the $0^{th} - 95^{th}$ percentile raw IP flows when sorted on the total α-time. The highest frequency occurs for 60 seconds, which means most α-flows have only one α-interval. As our algorithm for filtering out α-flows requires the amount of bytes sent in a minute to be greater than a threshold (1 GB in this analysis), if a flow lasts 70 seconds, its rate in the second minute could have been high but because this bytes threshold is not crossed, it is not recorded.

Not shown in the graph are the top 5 percentile of flows because these stretch out in time. The top flow lasted almost 3 hours. Online circuit setup is feasible for at least 1.6% of these flows if we use a factor of ten for the flow duration relative to the 1-minute circuit setup delay. But mechanisms are needed to predict which new incoming flows will last this long. Combining this problem with the finding that the most α-flows are relatively short-lived, the offline identification, circuit setup and PBR configuration design for HNTES seems more appropriate. This confirms our first hypothesis in Section IV.

TABLE I: Notation

Set symbol	Description	Number of elements	Elements of a set	Attributes of an element					
				Identifier	α-bytes	Start and end time	α-time	Number of reports /raw IP flows	Ingress and egress router ID pair
$\mathbf{R}_i$ $1 \leq i \leq \tau$	Set of NetFlow reports	m_i	$\mathbf{r}_{ij}$ $1 \leq j \leq m_i$	ω_{ij}	β_{ij}	(s_{ij}, e_{ij})	NA	NA	NA
$\mathbf{F}_i$ $1 \leq i \leq \tau$	Set of raw IP flows	n_i	$\mathbf{f}_{ik}$ $1 \leq k \leq n_i$	γ_{ik}	δ_{ik}	NA	ξ_{ik}	c_{ik}	$(\mathfrak{R}^I_{ik}, \mathfrak{R}^E_{ik})$
$\mathbf{P}_i$ $1 \leq i \leq \tau$	Set of prefix flows	d_i	$\mathbf{p}_{il}$ $1 \leq l \leq d_i$	ζ_{il}	η_{il}	NA	μ_{il}	g_{il}	$(\mathfrak{R}^I_{il}, \mathfrak{R}^E_{il})$

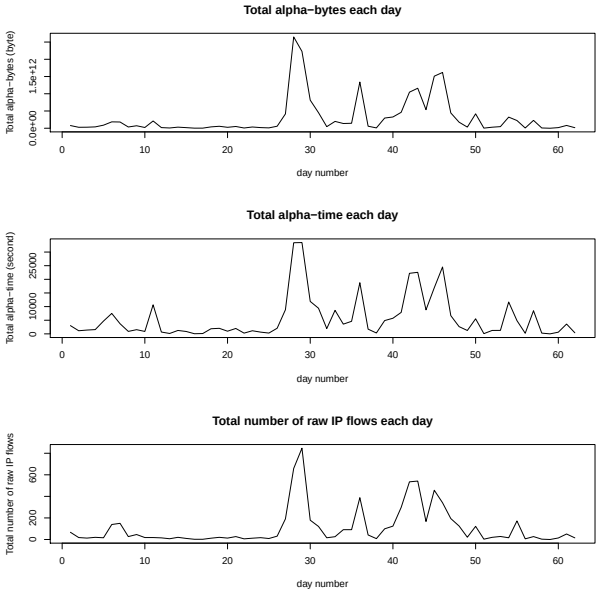

Fig. 1: Total number, α-bytes, and α-time (sec), of raw IP flows on a per-day basis

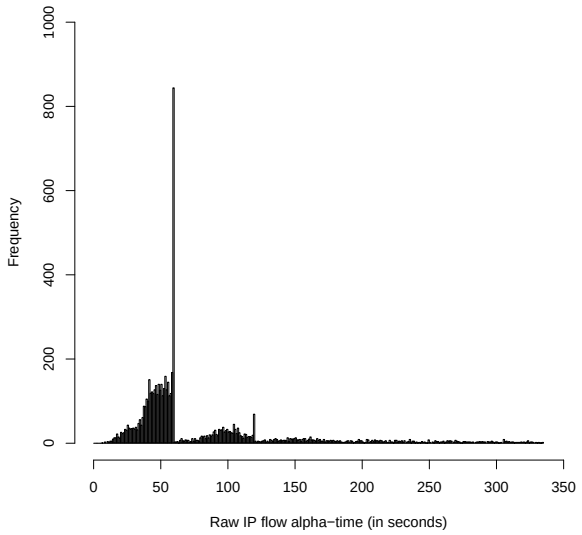

Fig. 2: A histogram of the total α-time of raw IP flows (0^{th} – 95^{th} percentile)

The total number of prefix flows across all 62 days is 60 for the /24 flows and 501 for the /32 flows. While these are fairly small numbers, the persistency measure becomes important before accepting our second hypothesis from Section IV. Fig. 3 shows that the number of prefix flows added per day, especially for the /24, decreases with time. On day 1, there were ten /24 new prefix flows but after day 41, there were only 0 or 1 new flows. The implication is we can keep adding /24 prefix

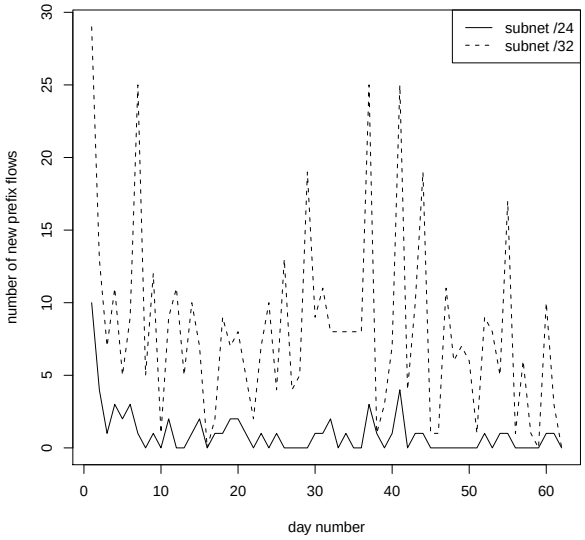

Fig. 3: The number of new prefix flows per day

flow identifiers to the PBR table without much concern for a growth in this table size. The cost of using /24 identifiers will be quantified in a future paper. General-purpose flows whose identifiers fall within these aggregated prefix identifiers will be redirected to the virtual circuits where they will face the adverse effects of α-flows.

To use provisioned circuits and offline configured PBRs, there should be some persistency of prefix flows. A histogram of the number of days in which each prefix flow appeared is plotted in Figure 4 for both /24 and /32 prefix flows. The maximum number of days (out of 62) in which a /24 prefix flow appeared is 33. 16.7% of /24 prefix flows appeared more than 15 days (which is a quarter of the observation period).

Data for the /24 and /32 prefix flows were sorted based on total α-bytes (where the total was computed across all 62 days for each prefix flow), per-day α-time, and per-day number of constituent raw IP flows. The data for 12 flows (/24 and /32) at 6 different percentiles is presented in Table II. For example, the data shown for the 100^{th} percentile flow corresponds to the top flow in each category, while the data shown for the 50^{th} percentile flow corresponds to the median flow in each category.

The prefix flow that had the maximum amount of total data transferred over 62 days, i.e. 11.9 TB, had constituent α-flows in 13 out of the 62 days, while the 80^{th} percentile flow when prefix flows were sorted by the maximum α-time per day had constituent α-flows in 33 out of the 62 days. On its worst day, this prefix flow had constituent raw IP flows with α-intervals that totaled 28.4 minutes. The worst prefix flow, in terms of

TABLE II: Data for particular flows corresponding to the percentiles
(the max/day and total numbers are over 62 days)

	Percentiles	100%		90%		80%		70%		60%		50%	
	Subnet length	/24	/32	/24	/32	/24	/32	/24	/32	/24	/32	/24	/32
Sorted on total α-bytes	max(α-bytes/day) (TB)	2.6	0.4588	0.18	0.0092	0.12	0.0046	0.0095	0.0045	0.0042	0.0029	0.0184	0.0022
	total bytes (TB)	11.87	3.22	0.3843	0.03	0.124	0.00815	0.07	0.0045	0.0309	0.0029	0.0184	0.0022
	no. of occurrences	13	12	23	8	3	2	23	1	17	1	1	1
Sorted on max(α-time/day)	max(α-time/day) (min)	540.6	278.2	124.9	9.0	28.4	3.0	12.85	1.8	7.69	1.0	5.38	1.0
	total α-time (min)	2974.4	470.5	130.8	9.0	172.9	3.0	14.1	7.3	7.69	1.0	5.38	1.3
	no. of occurrences	13	6	2	1	33	1	2	8	1	1	1	2
Sorted on no. of raw IP flows/day	max no. per day	649	221	67	9	14	4	9	2	6	2	4	1
	no. of occurrences	13	6	5	1	1	2	4	1	4	1	2	1

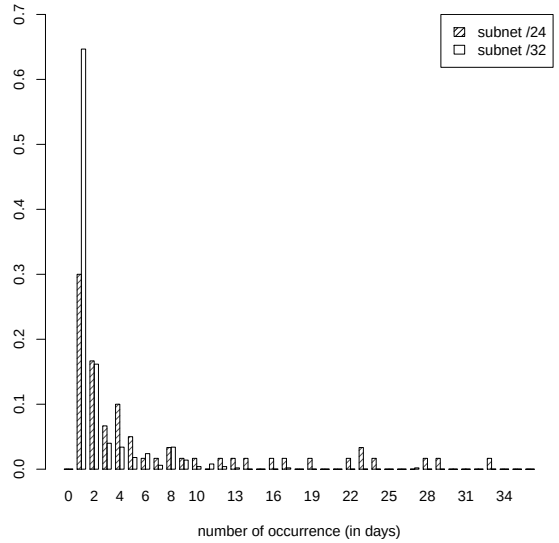

Fig. 4: Histogram of the number of days in which each prefix flow appeared

having constituent raw IP flows with the most per-day α-time, had α-flows occurring in 9 out of the 24 hours in the day (i.e., 540 mins). The final category was created by sorting on the per-day number of constituent raw IP flows. The prefix flow with the maximum value here had constituent raw IP flows in 13 days out of 62. The maximum value among these 13 for the per-day number of constituent raw IP flows was 649.

VI. Conclusions

A certain type of flows, referred to as α-flows, which generate a large number of bytes that exceeds some high threshold within fixed intervals (e.g., 1 min) are known to dominate other flows adding to the burstiness of the traffic, and are typically generated by large file transfers across paths with high bottleneck link rates. There is an interest in the scientific community to identify these flows and redirect them off the IP-routed network to reduce their negative impact on general-purpose flows. Toward this goal, designs for a hybrid network traffic engineering system that would identify such flows from NetFlow data and redirect them to a virtual-circuit network are considered. To determine whether these actions should be online or offline, we undertook traffic analysis of ESnet NetFlow records. Our findings are that α-flows identified by their 5-tuple (referred to as raw IP flows) are short-lived (80% are 2 minutes or less) making the use of online circuits with its high cost of setup delay infeasible. But to redirect prefix flows (identified by /32 or /24 source and destination IP addresses) to provisioned virtual circuits, persistency is required. For the observed period, 70% of the /24 prefix flows occur more than once.

VII. Acknowledgment

The University of Virginia portion of this work was supported by the DOE grant DE-SC002350, and NSF grants OCI-1038058 and OCI-1127340.

References

[1] ESnet. [Online]. Available: http://www.es.net/
[2] Internet2. [Online]. Available: http://www.internet2.edu/
[3] W. E. Johnston. The Science Data Network (SDN). [Online]. Available: http://www.es.net/ESnet4/ESnet4-Networking-for-the-Future-of-Science-2008-02-09-FORTH-Crete.ppt
[4] M. Veeraraghavan, M. Karol, and G. Clapp, "Optical dynamic circuit services," *Communications Magazine, IEEE*, vol. 48, no. 11, pp. 109 –117, november 2010.
[5] J. Wallerich, H. Dreger, A. Feldmann, B. Krishnamurthy, and W. Willinger, "A methodology for studying persistency aspects of internet flows," *ACM SIGCOMM Communication Review*, vol. 35, no. 2, 2005.
[6] Netflow. [Online]. Available: http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html
[7] Kun-chan Lan and John Heidemann, "A measurement study of correlations of internet flow characteristics," *Computer Networks*, vol. 50, no. 1, pp. 46–62, 2006.
[8] S. Sarvotham, R. Riedi, and R. Baraniuk, "Connection-level analysis and modeling of nework traffic," in *ACM SIGCOMM Internet Measurement Workshop 2001*, November 2001, pp. 99–104.
[9] K. Papagiannaki, N. Taft, S. Bhattacharyya, P. Thiran, K. Salamatian, and C. Diot, "A pragmatic definition of elephants in internet backbone traffic," in *IMW '02 Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, 2002, pp. 175–176.
[10] N. Brownlee and K. Claffy, "Understanding Internet traffic streams: dragonflies and tortoises," *Communications Magazine, IEEE*, vol. 40, no. 10, pp. 110 – 117, oct 2002.
[11] T. T. T. Nguyen and G. J. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.
[12] X. Xiao, A. Hannan, and B. Bailey, "Traffic engineering with mpls in the internet," *IEEE Network Magazine*, vol. 14, pp. 28–33, Mar. 2000.
[13] flow-tools. [Online]. Available: http://www.splintered.net/sw/flow-tools/docs/flow-tools.html
[14] The R Project for Statistical Computing. [Online]. Available: http://www.r-project.org/