J Netw Syst Manage (2016) 24:1–33
DOI 10.1007/s10922-014-9336-2

A Network Management System for Handling Scientific Data Flows

Zhenzhen Yan · Chris Tracy · Malathi Veeraraghavan · Tian Jin · Zhengyang Liu

Received: 13 January 2014 / Revised: 6 September 2014 / Accepted: 26 September 2014 /
Published online: 11 October 2014

Abstract Large scientific data transfers often occur at high rates causing increased burstiness in Internet traffic. To limit the adverse effects of these high-rate large-sized flows, which are referred to as α flows, on delay-sensitive audio/video flows, a network management system called Alpha Flow Traffic Engineering System (AFTES) is proposed for intra-domain traffic engineering. An offline approach is used in which AFTES analyzes NetFlow records collected by routers, extracts source–destination address prefixes of α flows, and uses these prefixes to configure firewall filters at ingress routers of a provider's network to redirect future α flows to

The UVA portion was supported by NSF grants OCI-1127340, CNS-1116081, ACI-1340910, CNS-1405171 and U.S. DOE grants DE-SC0002350 and DE-SC0007341. The ESnet portion was supported by the Director, Office of Science, Office of Basic Energy Sciences, of the U.S. DOE under Contract No. DE-AC02-05CH11231. This research used resources of the ESnet ANI Testbed, which is supported by the Office of Science of the U.S. DOE under contract DE-AC02-05CH11231, funded through the American Recovery and Reinvestment Act of 2009.

Z. Yan · M. Veeraraghavan (✉)
Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22904-4743, USA
e-mail: mv5g@virginia.edu

Z. Yan
e-mail: zy4d@virginia.edu

C. Tracy
Energy Sciences Network (ESnet), Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
e-mail: ctracy@es.net

T. Jin · Z. Liu
Department of Computer Science, University of Virginia, Charlottesville, VA 22904-4740, USA
e-mail: tj3sr@virginia.edu

Z. Liu
e-mail: zl4ef@virginia.edu

traffic-engineered paths and isolated queues. The effectiveness of this scheme was evaluated through an analysis of 7 months of NetFlow data obtained from an ESnet router. For this data set, 91 % of bytes generated by α flows during high-rate intervals would have been directed had AFTES been deployed. The negative aspect of using address prefixes in firewall filters, i.e., the redirection of β flows to α-flow paths/queues, was also quantified.

Keywords NetFlow traffic analysis · Elephant flows · Scientific computing · Research and education networks (RENs) · MPLS · Virtual circuits

1 Introduction

Scientific computing applications in fields such as high-energy physics, climate science, genomics, etc., generate large (tera- to peta-byte sized) data sets [1]. To transfer these data sets at high speeds, scientific users often invest in high-end computing clusters with disk arrays, parallel file systems, and high-speed access links. Usage logs collected at these data-transfer servers show that some transfers occurred at a significant fraction of link capacity, e.g., 4 Gbps[1] on 10 Gbps links [2]. New TCP variants such as H-TCP [3] are used to generate such high rates for single flows. The high-rate large-sized transfers, referred to as α flows, are the primary source of burstiness in IP traffic [4].

Core research-and-education network providers, such as US Department of Energy (DOE)'s Energy Sciences Network (ESnet) [5], have identified such α flows as having adverse effects on general-purpose (β) flows. As α flows cause burstiness, audio/video applications experience packet delay variance (jitter) and a corresponding degradation in performance. Such degradations in performance result in trouble tickets that add to a provider's operational costs. To address these costs, DOE has supported research on traffic engineering systems. We propose one such system to (i) identify high-rate large-sized data-transfer flows from the packet traffic entering ingress routers of a provider's network, (ii) control the path taken by these flows by establishing intra-domain virtual circuits (traffic engineering), and (iii) isolate packets from these flows into separate virtual queues to reduce their effects on general-purpose flows [6–8].

The first task of the traffic-engineering system listed above, which is to identify high-rate large-sized flows from the packet traffic entering a provider's network, is the problem statement addressed in this work.

Basis for solution approach A seemingly simple solution is to modify end-user applications, such as GridFTP [9], to signal a provider's network with a control-plane message before initiating any high-rate large-sized transfers. Such a message would negate the need for automatic α flow identification systems within a provider's network. This solution was attempted in projects such as Lambdastation [10], Terapaths [11], and CHEETAH [12], but practical difficulties of application

[1] Unlike in residential networks where link capacity is the bottleneck, in scientific laboratories, the bottleneck is the end-system computing/disk resource, not links.

upgrades and adoption by users hindered its deployment. This led us to pursue an intra-domain traffic engineering solution because deployment of such a system would be entirely within a provider's control. Such a solution does not preclude a parallel technology adoption effort of the end-application signaled approach.

For provider networks to automatically identify α flows, we start by examining the available features in current-day IP routers. Unfortunately, routers do not have built-in mechanisms to determine the rate and size of a flow (where a "flow" is identified by the 5-tuple: source and destination IP addresses, source and destination transport-layer port numbers, and protocol type). Next, we considered a port-mirroring mechanism in which IP routers could be configured to make and transmit copies of packets to a port that is connected to an external server; the latter could then be used to execute a flow-based rate/size analysis for α-flow identification. However such a mechanism was deemed unscalable for the high link rates (10–100 Gbps) of provider networks.

After concluding that there are no built-in mechanisms for flow rate/size computation within routers, and that port-mirroring is infeasible, we looked for other mechanisms that could be exploited. Our finding is that NetFlow, a feature supported in provider-scale IP routers, can be used to solve our problem [13]. The NetFlow feature allows routers to collect information for a sampled set of packets, which is then exported, in the form of NetFlow records, to an external `NetFlow Collector`. In current-day installations, NetFlow records are exported on a coarse time granularity, which is on the order of minutes to hours. An analysis of the NetFlow data showed that it was not possible to accurately predict the duration and size of an α flow by observing the first few NetFlow records corresponding to a live (online) flow. Any traffic-engineering/flow isolation actions taken on the presumption of a flow being an α flow may be futile in that the flow could end even before the router-configuration actions for traffic-engineering/flow-isolation were completed. Therefore, we developed an offline mechanism in which NetFlow records from completed flows are analyzed, and information extracted from this analysis is used to configure routers to identify future α flows for traffic-engineering/flow-isolation.

Solution approach We propose a network management system called *Alpha Flow Traffic Engineering System (AFTES)* that would be run on a server external to the routers.[2] AFTES would obtain NetFlow records from the NetFlow collector, and store the source–destination address prefixes of already completed α flows. These prefixes are used to configure firewall filters at ingress routers so that future α flows between the same source/destination subnets will get redirected to traffic-engineered, QoS-controlled paths. A persistence measure is used to delete address prefix entries from the firewall filters for which no α flows are observed over an aging interval. This AFTES design would be effective if the following hypothesis is true.

[2] This system was proposed in our conference papers [6, 7]. It was called Hybrid Network Traffic Engineering System (HNTES). The name was changed to AFTES to reflect its functionality more accurately.

Hypothesis Most high-speed data transfer nodes have static IP addresses, and α flows are created repeatedly between the same source–destination subnets. The basis for this hypothesis is that scientists typically execute their simulations on the same supercomputing centers, and hence we expect them to transfer data between the same two clusters. If the hypothesis is true, the offline prefix identifier based AFTES scheme will be effective in identifying and directing α flows to traffic-engineered, QoS-controlled paths. We carried out traffic analysis of NetFlow records collected from ESnet to test this hypothesis.

Findings Our data analysis showed that if AFTES had been deployed at the start of the 7-month period for which NetFlow records were analyzed, 91 % of bytes from α flows that occurred in high-rate intervals would have been sent to traffic engineered, QoS-controlled paths. This validated the hypothesis for the tested ESnet router/time period; to enable other researchers to test this hypothesis with data procured from other providers,[3] our software has been made available on a public Web site [14].

Our second finding relates to quantifying the cost of redirecting flows based on source–destination address prefixes rather than on the more constrained 5-tuple identifiers (which includes source and destination IP addresses, source and destination port numbers, and protocol type). The effect of this design choice is that β (non-α) flows that share the same source–destination address prefixes as α flows will get redirected to the α-flow paths and queues. However, our data analysis shows that a majority of such β-flow packets are from file-transfer applications and not delay-sensitive interactive applications. This is likely because most scientific computing/data transfer nodes are located in separate subnets from those used to connect general-purpose user hosts.

Novelty Three novel ideas form the basis for AFTES: (1) Apply Quality-of-Service (QoS) controls to file transfers, not interactive audio/video flows; (2) Use intra-domain, not end-to-end, virtual circuits; and (3) Exploit knowledge of human behavior and end system capabilities in developing a flow classification algorithm.

When Asynchronous Transfer Mode (ATM) [15] and Integrated Services (IntServ) [16] technologies were developed, virtual circuit services were considered for interactive audio/video flows. However, because the number of such flows is large, and per-flow QoS controls can only be applied to small numbers of flows in most routers, these virtual-circuit solutions were not as widely deployed as originally envisioned. Instead, IP-routed networks were simply over-provisioned. Such over-provisioning is usually sufficient to keep delay/jitter low for interactive audio/video flows. However, α flows can, in short intervals, quickly fill up router buffers. Scientific data-transfer servers that generate α flows typically have 10 Gbps network interface cards, and can hence burst out packets at that rate, which was also the rate of core network links. Such bursts can cause observable quality reductions in interactive audio/video flows (see Sect. 2). Our novel approach is to apply QoS

[3] It is difficult to procure NetFlow data from providers for privacy reasons and therefore we could not test our hypothesis with other providers' data. But we did extend our prior 2-month analysis [7] to a 7-month analysis presented in this paper.

controls to α flows, which are fewer in number than audio/video flows, and can hence be handled by today's routers.

The second novel concept proposes using intra-domain virtual circuits, while most past work requires the use of end-to-end virtual circuits. If QoS metrics, such as end-to-end delay, need to be guaranteed, then virtual circuits must extend end-to-end. However, delay guarantees are not required for α flows, and technologies exist for creating hybrid end-to-end paths, in which some segments of a path are IP-routed while other segments are virtual circuits. The use of intra-domain virtual circuits allows a single provider to unilaterally deploy AFTES for better traffic engineering. This makes the adoption of AFTES more likely.

The third novel concept is our exploitation of human-behavior knowledge and end-system capabilities in the AFTES design. Other work on flow classification algorithms, as presented in Sect. 3.2, proposed the use of payload-based, port-based or machine learning techniques, whereas our AFTES solution is based on pragmatic observations about who creates α flows, and the type of end systems required to create such flows.

Contributions The AFTES design and evaluation are the main contributions of the work. The AFTES design is a pragmatic solution that solves a technical problem (reducing the adverse effects of α flows on other flows), while taking into account constraints of today's routers and difficulties of deploying inter-domain solutions. The evaluation showed that the AFTES design is a viable solution for deployment.

Significance For practitioners, the AFTES prototype, being developed for potential deployment in ESnet, could be of direct benefit. The design strategy that requires only NetFlow records, collection of which is already supported by most providers, lowers the barrier to its acceptance by providers. As mentioned earlier, the intra-domain nature of the AFTES solution is also attractive to practitioners.

Section 2 describes the motivating factors for this work. Section 3 provides background information and reviews related work. The system architecture is described in Sect. 4. Section 5 describes the proposed offline α flow identification mechanism. An evaluation of this mechanism through NetFlow data analysis of ESnet traffic is presented in Sect. 6. The paper is concluded in Sect. 7.

2 Motivation

This section motivates the reason for needing to isolate α flows to their own traffic-engineered paths, and a shorter version of this motivation was presented in a 2013 conference publication [6]. First, an example of measured traffic levels, reaching close to the link capacity of an ESnet router interface, is provided. Next, we describe an experiment to illustrate that a single α flow can cause such increases in traffic levels, and that such increases can adversely impact packet delays in other flows.

Figure 1 plots Simple Network Management Protocol (SNMP) link utilization of an ESnet router interface. This data was collected on Jan. 16, 2013. The traffic level reached above 9 Gbps (the link's capacity was 10 Gbps). These sudden increases in traffic levels are most commonly caused by a few α flows or even a single flow. For the particular example shown in Fig. 1, we extracted the NetFlow records for this

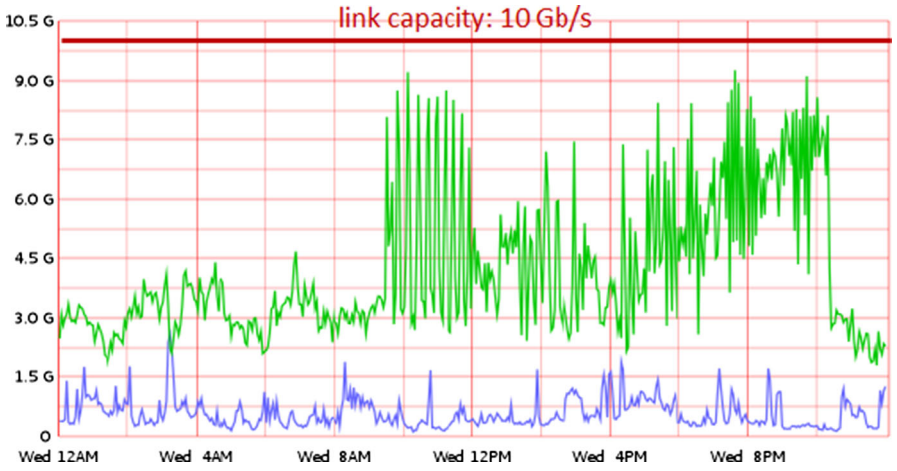

Fig. 1 Utilization (b/s) of a 10 Gbps ESnet router interface observed on Jan. 16, 2013; the *green line*, showing bursts reaching over 9 Gbps, is the outgoing traffic from the ESnet router to a peering REN, while the lower-load *blue line* is the incoming traffic on the same interface [6]

time period and found the source and destination of the flow that caused these spikes. Research-and-Education Networks (RENs) such as ESnet are engineered to operate at 20–25 % loads in order to absorb α-flow spikes [17]. Typically, traffic in both directions of the link shown in Fig. 1 have loads similar to that of the blue line (incoming direction) [18]. Inspite of the capacity headroom, α flows can cause adverse effects as demonstrated next.

We conducted an experiment on a 10 Gbps ESnet-run IP-routed metropolitan-area experimental network [8]. Its high-performance servers, each equipped with 10 Gbps network interface cards, were capable of sourcing/sinking data at multiple Gbps. The experiment consisted of initiating the following flows: (1) a UDP flow generated using the `nuttcp` application [19] (rate of the flow was configured to be 3 Gbps), (2) a TCP flow was started at 53 seconds using the `nuttcp` application, and (3) the `ping` application was started with the appropriate arguments to send a periodic ICMP Echo-Request message. The graphs of Fig. 2 illustrate these three flows. In the 1-queue configuration, packets from all three flows were buffered in the same output queue. The throughput of the TCP flow was more than 6 Gbps, but the delays experienced by the ICMP packets of the ping application increased from 2.3 ms (before $t = 53$ when the TCP flow was started) to 60.6 ms (until $t = 152$ when the TCP flow was terminated). In the 2-queues configuration, the router directed packets from the UDP flow and the ICMP packets generated by the ping application to one virtual queue and the TCP-flow packets to a separate virtual queue on the contending egress interface. Weighted-fair queueing (WFQ) was used for packet scheduling. The rate allocation was 40 and 60 % for the two queues, respectively. The scheduler was configured to function in work-conserving mode, i.e., bandwidth partitioning was not strict. This mode of operation allowed the TCP-flow packets to be served at 6 Gbps, even as it maintained the delay experienced by the ICMP packets of the ping flow to the low value of 2.3 ms. This experiment

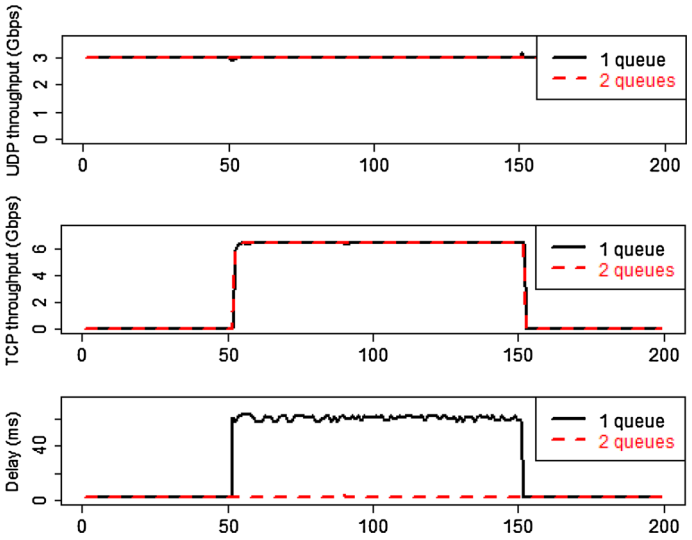

Fig. 2 The x-axis is time measured in seconds; the *top graph* shows that the UDP-flow rate is 3 Gbps in both the 1-queue and 2-queues configurations; the *middle graph* shows the TCP flow throughput; the *bottom graph* shows the delays experienced in the ping application [8]

illustrates the need for isolating α flows to a separate queue in order to reduce the negative impact of such flows on real-time flows.

3 Background and Related Work

3.1 Background

Terminology Two terms, *5-tuple flow* and *prefix flow* [20], are used in this paper. A *5-tuple flow* consists of all packets arriving with the same 5-tuple values {source IP address, destination IP address, source port number, destination port number, protocol type} with no consecutive inter-packet gaps greater than some fixed time threshold. The fixed time threshold phrase is required because TCP and UDP port numbers for the same source–destination hosts are reused at some point in time. A *prefix flow* consists of all packets arriving within an aggregation interval (some fixed duration) that have the same source and destination address prefixes (e.g., /24 prefixes).

ESnet ESnet is a US-wide core (backbone) high-speed REN that offers IP-routed and dynamic virtual circuit services to DOE national laboratories, such as Argonne National Laboratory, Brookhaven National Laboratory, and several others [5]. As the DOE national laboratories conduct scientific research in many disciplines such as high-energy physics, α flows created by the movement of scientific datasets are observed on ESnet router interfaces (e.g., see Fig. 1). As the NetFlow data analyzed

in this work was collected in 2011,[4] the corresponding ESnet network topology is shown in the 2010 network map [21]. There were 75 routers in total, with 42 routers located in customer premises as provider edge (PE) routers, and the remaining routers were used in the core backbone (routers are located in cities such as Houston, Atlanta, etc.) and in three metro-area rings in Chicago, Northern California, and New York. All backbone links, and links from major PE routers to core routers were 10 Gbps Ethernet. ESnet peers with other US backbone RENs such as Internet2, and international RENs such as GEANT2, and with commercial peers and provider networks. Several of these providers offer dynamic virtual circuit services and DiffServ based services [22–26], which are briefly reviewed next.

NetFlow NetFlow v5, originally proposed by Cisco, is being replaced by NetFlow v9 [27] and IPFIX [28]. However, the ESnet4 (Juniper) routers from which data was collected for this study implemented NetFlow v5, and hence we provide a brief review of NetFlow v5 here. NetFlow enables IP routers to collect information from packet headers (sampled or unsampled), and save information as flow records. Each IP router's NetFlow system maintains a running set of flow records. Each record includes the arrival timestamps of the first and last packets, number of observed packets, and the total flow size in bytes. At the end of each *active timeout interval*, a NetFlow record is created. Thus, multiple NetFlow records may be created for a flow. These stored flow records are sent from the IP router's `NetFlow exporter` to a `NetFlow collector` (a process running on an external host). Processing and maintaining volumes of NetFlow data can be both computationally and storage intensive, especially for routers with high-speed links.

Firewall filters This feature of routers will be leveraged in our system design for flow redirection. The firewall filter rules are consulted prior to the default IP routing tables to determine how to forward incoming packets. Unlike with the default IP routing table in which only the destination address is used to determine the output link, firewall filter rules can be set to match any subset of the 5-tuple identifiers. This feature is used by AFTES.

DiffServ, MPLS, and hybrid paths Differentiated Services (DiffServ) [29] is a technique for handling multiple classes of service, with packets from different classes being buffered in different virtual queues. DiffServ Code Point (a field of the IP header) is used to identify packets from different classes. Weighted fair queuing (WFQ) and priority queueing (PQ) can be configured to enable fair sharing of link capacity between the multiple classes.

Multi-Protocol Label Switching (MPLS) is a virtual-circuit networking technology. Virtual circuits have to be provisioned across the network before packets can be forwarded on the labels carried in their MPLS headers. QoS controls, such as policing and scheduling, can be based on MPLS labels.

Hybrid paths consist of IP-routed segments and virtual circuits (VC). IP routers have built-in MPLS switching engines, and support the so-called "Layer-3 MPLS" capability, which allows for the router to forward a filtered set of IP packets to a MPLS virtual circuit (label switched path).

[4] In 2012, ESnet had a major upgrade with all backbone link capacities increased to 100 Gbps.

Arguably, α flows could be handled using Diffserv instead of MPLS since the main goal is to isolate α flows from other general-purpose flows. In a complementary effort [8], we showed that α flows can be separated into their own virtual queue, and with WFQ and PQ, both goals of keeping packet delays low for interactive audio/video flows and throughput high for α flows can be achieved. However, the reason for choosing MPLS for AFTES is that the virtual-circuit setup phase offers the opportunity for a provider's network management system to choose paths for the α flows that are possibly different from the default IP-routed paths for load balancing or other reasons.

The above discussion is included here to provide readers a holistic view of AFTES, but this particular paper focuses on just the first task, identifying high-rate large-sized flows. Our other paper [8] focuses on the traffic-engineering and packet isolation tasks of AFTES.

3.2 Related Work

Prior papers on flow classification, traffic engineering, methods for handling elephant flows, and OpenFlow/SDN are reviewed briefly. We also compare this paper with our prior related conference publications.

Flow classification Lan and Heidemann [30] identify four dimensions of flows: size (bytes), rate, duration, and burstiness. Flows are classified as elephants or mice based on size, cheetahs or snails based on their rates, tortoises or dragonflies based on their duration, and porcupines or stingrays based on their burstiness. Further, four methods are used to define the thresholds between the two classes for each of these dimensions. In the first method, the numbers corresponding to the mean + 3 standard deviations of the sampled data are used as the thresholds, based on earlier work by Sarvotham et al. [4]. In the second method, the top 1 % of all flows is used to set the thresholds. In the third method, the thresholds are cutoff points in a heavy-tailed distribution where the cutoff is determined using the *aest* method [31]. In the fourth method, the thresholds are set so that 50 % of all traffic is carried by the heavy-hitters (elephants, cheetahs, etc.). The goals of the Lan and Heidemann study were to characterize flows along the four dimensions and to study correlations between different flow types, while the purpose of Sarvotham's study was to create a framework for modeling network traffic.

Other papers of relevance are on flow classification algorithms [32, 33]. Flow classification methods are grouped in [33] into three categories: (1) port based, (2) payload based, and (3) flow statistics based. Scientific data transfer applications, such as GridFTP, use ephemeral ports and control-plane packet encryption, which render port-based and payload-based classification methods unsuitable for our purposes. Our solution falls in the last category of using flow statistics. In general, as noted in [33], machine learning techniques are required for flow-statistics based classification schemes because of the large size of datasets and multi-dimensional flow parameters. However, our solution does not require machine learning techniques because our hypothesis about the repetitive patterns of α flows was validated for scientific data transfers.

Traffic engineering Survey papers [34, 35] describe different approaches for traffic engineering. Specifically, our approach fits the "tactical traffic engineering" category [34] rather than "strategic traffic engineering."

Identifying and handling elephant flows Papagiannaki et al. [36] define elephants to be flows that are the major contributors to network load, i.e., flows whose bandwidth exceeds a threshold value and exhibit persistency. The threshold is a relative value such that "a flow is characterized as an elephant, only if it is located in the tail of the flow bandwidth distribution [36]." In contrast, our definition of α flows uses a fixed threshold for bytes observed within a fixed duration. Only flows that exceed the threshold have the potential to cause packet delays for real-time flows as discussed further in Sect. 6. Using relative values of bandwidth, as is done in the paper by Papagiannaki et al. [36], may result in the classification of certain flows as elephant or α flows, even if their rates/sizes are not large enough to have adverse effects on other flows. Therefore our work uses absolute thresholds. These threshold values are based on an analysis of actual scientific data transfers logged by GridFTP servers [2].

For their analysis purposes, Wallerich et. al [20] classified a flow as an elephant if in any time bin (which was chosen to be 1 min) in its lifetime, the number of bytes sent is in the top ranks of all flows. It is thus also a relative classification scheme and hence differs from our work for the same reason described above. Other such papers are surveyed by Callado, et al., in a 2009 paper [37].

Kamiyama and Mori [38] propose a short-timeout method to identify high-rate flows and elephant (large) flows [39] with low false-positive and false-negative rates. Zhang et al. [40] proposed a Bayesian single sampling method to identify high-rate flows. Duffield et al. [41] had a goal of finding information about flows in unsampled packets using information in sampled packets.

An online technique for identifying large IP flows based on prediction for redirection to optical lightpaths was proposed by Fioreze et al. [42]. A relationship is observed between flow size and bits per second, duration, and packets per sec. Bypass to circuits have been used for other applications such as traffic matrix computation [43].

There are several papers proposing methods for identifying large flows or high-rate flows with new router hardware. These include ElephantTrap [44], RATE [45], CATE [46], an FPGA-based cache solution [47], and a real-time detector for Grid bulk-data transfer flows on 1 Gbps links [48]. Also Hohn and Veitch [49] proposed a scheme for finding the spectral density, distribution of the number of packets per flow, and showed why alternate sampling techniques were needed to obtain this second-order statistic about flows. Given our focus on designing network management systems and not new router hardware, our scheme relies on the built-in NetFlow system supported in most deployed provider routers.

Hybrid packet-switched and optical circuit-switched networks have been proposed for datacenters [50–52]. Different techniques are proposed in these papers for determining which flows should be directed to optical circuits. For example, in HELIOS [51], elephant flows are identified using a static flow-rate cutoff (15 Mbps) for automatic redirection to optical circuits.

OpenFlow/SDN OpenFlow [53] and Software Defined Networing (SDN) [54] are enabling easier access to flow tables, and allow for more flexible traffic engineering. As OpenFlow switches and routers are currently being deployed, the networking community is developing new tools to leverage these capabilities. For example, Qazi et al. [55] proposed a machine-learning technique for use in an SDN to identify the application types of flows. Wang et al. [56] proposed the use of application-aware SDN for big-data applications. As mentioned in Sect. 1, difficulties such as application upgrades and adoption by users hindered the deployment of application-aware traffic engineering solutions in core provider networks, and therefore our solution is based on in-network α-flow identification.

Comparison with our prior papers As mentioned in Sect. 1, we have published two conference papers on HNTES [6, 7]. In our 2012 paper [7], we introduced HNTES and presented preliminary analysis of 2-months NetFlow records. In our 2013 paper [6], we presented a high-level comparison of the effectiveness and cost of HNTES by analyzing data from 4 ESnet routers. In contrast, this manuscript presents an in-depth evaluation of the effectiveness and cost of HNTES showing more detailed graphs and tables. Next, this manuscript has a detailed characterization of α prefix flows and a data-door analysis of the source and destination of α flows. Both these characterizations are new contributions of this manuscript.

4 AFTES Overview

This section provides an architectural overview of how AFTES could be deployed in a provider network [6, 7]. Consider the example provider network shown in Fig. 3. Default IP-routed paths from router A to router C in the example provider network are shown with red dashed lines. AFTES is a network management software system that would be run on an external server as shown in Fig. 3. AFTES interacts with two external systems, a NetFlow collector, and an Inter-Domain Controller (IDC). The role of a NetFlow collector is explained in Sect. 3.1. The IDC is a virtual circuit scheduler and is used in RENs such as ESnet and Internet2 to support advance-reservations for virtual circuits (VCs) [25]. Both providers use MPLS to support their dynamic virtual circuit service offerings. AFTES leverages this service for α flows. The setup phase in VC networking offers the opportunity for traffic engineering α flows along paths distinct from the default IP-routed paths if needed (e.g., for load balancing).

AFTES operations are grouped into two phases:

α -flow address prefix identification AFTES procures NetFlow records from the NetFlow collector (as shown in Fig. 3), and extracts the source and destination IP address prefixes of α flows. Details regarding the thresholds used in determining which flows are α flows are provided in the next section.

In ESnet, we expect to run AFTES on a nightly basis, and therefore we use a per-day index i in creating a set of α prefix IDs $\mathbf{F}_i$, using the source–destination address prefixes of α flows observed during the day. To keep the set $\mathbf{F}_i$ from becoming too large (as this set determines the firewall filter rules that will be added to the routers), address prefix pairs for which α flows have not been observed within an aging

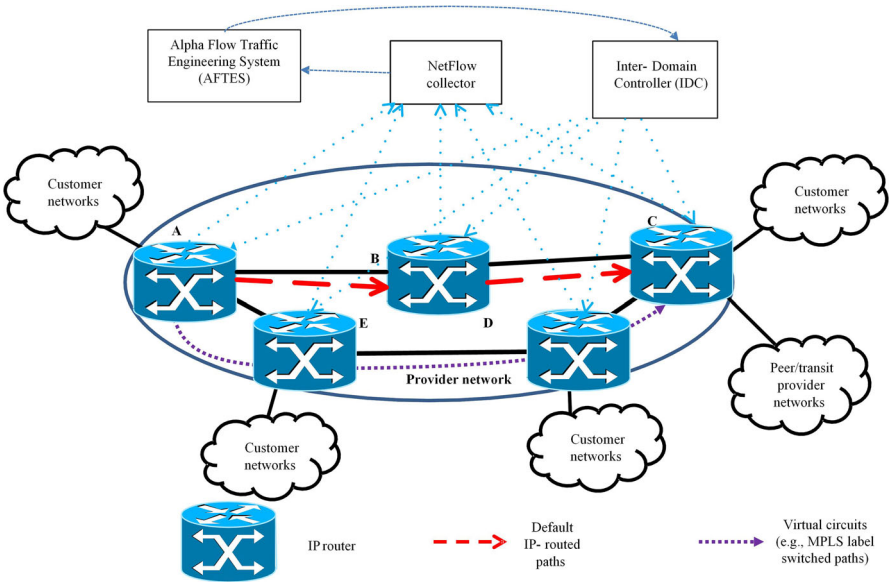

Fig. 3 Illustration of the role of Alpha Flow Traffic Engineering System (AFTES)

interval (e.g., 30 days) will be deleted. The reason for having an aging interval is to provide stability with respect to the firewall filter rules. This assumption about running AFTES on a nightly basis is a starting assumption. Based on observed repetition patterns, AFTES could be run more often. For example, if in a given provider's network, we observe that a source–destination pair performs multiple large-sized high-rate transfers within a day, and the cost of running AFTES and configuring routers is acceptable, then the frequency at which AFTES is executed can be increased. Also, an administrator can manually execute AFTES operations if significant unexpected changes are observed in network traffic.

Configuring routers for future α *-flow redirection* As noted in Sect. 1, the second and third tasks are to (ii) control the path taken by α flows by using virtual circuits, and (iii) isolate α flows into separate virtual queues to reduce their effects on general-purpose flows. Although these tasks are outside the scope of this paper as mentioned in Sect. 1, we provide a brief overview of solutions for these tasks.

To configure routers for future α-flow redirection, AFTES first determines the egress router E corresponding to an α prefix ID that is newly identified from NetFlow records obtained from ingress router I. It then checks to see whether an LSP already exists from ingress router I to egress router E. If it does, then AFTES communicates with the ingress router I to configure a new firewall filter rule, which will redirect future α flows whose prefix IDs match the newly identified α prefix ID to the existing LSP. If, on the other hand, AFTES finds that there is no existing LSP from ingress router I to egress router E, AFTES communicates with the IDC to initiate LSP provisioning. IDC includes the functionality to set firewall filter rules, and therefore AFTES can simply pass the α prefix ID to the IDC while requesting

LSP setup. The AFTES solution is offline in that α prefix IDs of completed flows are used to configure routers for future α-flow redirection. Therefore, there is no path switching in the middle of a flow, and hence out-of-sequence packets (which are adverse to TCP throughput) will not be caused by this traffic engineering solution.

With regards to *path selection* for the LSPs for α flows, one approach is to collect traffic statistics on the LSPs using SNMP. Xiao et al. proposed this approach as a means for obtaining the traffic matrix for a provider's network [57]. By provisioning LSPs with no bandwidth constraints between all ingress–egress router pairs, SNMP traffic statistics offer an easier way to obtain the traffic matrix when compared to methods that start with NetFlow records from sampled packets. From the SNMP traffic statistics, the IDC can run path computation methods, several of which are described in a survey paper [58]. Also survey papers on traffic engineering [34, 35] offer methods for load balancing, e.g., to minimize Maximum Link Utilization (MLU) [59].

With regards to *flow isolation*, we recommend configuring the router schedulers to operate in work-conserving mode, which means there is no strict partitioning of bandwidth among the various LSPs and IP-routed flows sharing an interface. In another paper [8], we demonstrated the negatives of policing α flows (since most of these flows use TCP) and proposed a no-policing, WFQ/PQ scheduling approach. Unlike circuit multiplexing techniques (e.g., TDM, WDM), virtual-circuit (VC) techniques such as MPLS separate the task of path selection/VC provisioning from the task of resource (bandwidth and buffer) allocation for QoS control [57]. In our paper [8], we showed the importance of creating multiple virtual queues so that delay-sensitive IP-routed packets are not held behind α-flow bursts. The work-conserving mode of resource sharing makes it less urgent to remove an LSP between an ingress–egress router pair if there are no firewall-filter rules directing flows with specified α prefix IDs to the LSP. Therefore, a delayed release approach can be implemented. LSPs are expected to be long-lasting as repeated α flows have been observed with the same prefix ID.

5 Characterizing α Flows

Terminology Table 1 first defines α *NetFlow records* as records whose total bytes exceeds a threshold H. Next, Table 1 defines an α *flow* as a 5-tuple flow (see Sect. 3.1 for definition) for which there is at least one α NetFlow record. Thus, an α flow may have some α NetFlow records and some non-α NetFlow records. An α flow can have one or more α-*intervals*, each of which is a time period in which the α flow sent packets whose aggregate size exceeded H bytes within a NetFlow active timeout period (denoted τ). Finally the source and destination address prefixes of α flows are referred to as α *prefix identifiers*.

The next three terms in Table 1 define α *prefix flows* and their characteristics, α- *bytes* and α-*time*. A prefix flow (as defined in Sect. 3.1) is an α prefix flow if its source/destination address prefix is an α prefix identifier. Two characteristics of an α prefix flow, α-bytes and α-time, are used to represent the aggregate bytes and total time observed in α-intervals of α flows that shared the prefix flow's identifier.

Table 1 Terminology

Term	Meaning
α NetFlow record	A NetFlow record in which the byte threshold H is exceeded
α flow	A 5-tuple flow (see Sect. 3.1 for definition) for which there is at least one α NetFlow record
α-interval of an α flow	Interval between first-packet and last-packet timestamps in each α NetFlow record of the α flow
α prefix identifiers (IDs)	Prefix identifiers (source/destination address prefixes) of α flows
α prefix flow	A prefix flow (see Sect. 3.1 for definition) in which all its packets belong to α flows that share its prefix identifier
α-bytes of an α prefix flow	Sum of bytes recorded in its constituent α NetFlow records
α-time of an α prefix flow	The total time within each aggregation interval in which at least one of the constituent α flows experienced an α-interval.

Notation Table 2 lists notation for parameters of α NetFlow records and α prefix flows, which are used to compute α-bytes and α-time for α prefix flows. In day i, the set of NetFlow records $\mathbf{R_i}$ has m_i α NetFlow records. Each record $\mathbf{r_{ij}}$ is itself a set consisting of several parameters, such as the number of bytes β_{ij}, source/destination address prefixes denoted as the identifier, ω_{ij}, and start time s_{ij} and end time e_{ij}, which represent the UTC timestamps of the first and last packets in the NetFlow record, respectively. The number of bytes in all NetFlow records in set $\mathbf{R_i}$ are lower-bounded by H, i.e.,

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

The next row in Table 2 represents the parameters of α prefix flows. On each day i, a set $\mathbf{P_i}$ of α prefix flows is created. Each element in this set is itself a set consisting of an identifier ζ_{il}, which is the source/destination address prefix pair, and other parameters as shown in Table 2.

Characteristics of α prefix flows Two characteristics, α-bytes and α-time, are defined for an α prefix flow. For α prefix flow $\mathbf{p}_{il}$, the α-bytes, η_{il} (see Table 2), is determined as follows:

$$\eta_{il} = \sum_{j=1}^{g_{il}} \beta_{ij},\ \text{s.t. }, j \epsilon \mathbf{J}_{il}, 1 \leq l \leq d_i \tag{2}$$

where $\mathbf{J}_{il} = \{j_1, j_2, \cdots, j_{g_{il}}\}$, a set of indices selected from record set $\mathbf{R}_i$ such that the source/destination address prefixes in identifiers ω_{ij_a} are equal to ζ_{il} for $1 \leq a \leq g_{il}$ and $1 \leq j_a \leq m_i$ (see Table 2).

To determine α-time of $\mathbf{p}_{il}$, the following procedure is used. If there are multiple α flows that share the prefix ID of an α prefix flow $\mathbf{p}_{il}$, these flows could have overlapping α-intervals. Such overlapping intervals should be counted only once. The α-intervals of constituent α flows in an α 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-

Table 2 Sets created for the ith aggregation interval

Symbol	Set	No. of elements	Elements of the set	Attributes of an element				
				Identifier	α-bytes	Start and end time	α-time	No. of α NetFlow records aggregated
$\mathbf{R}_i$	Set of α NetFlow records	m_i	$\mathbf{r}_{ij}$	ω_{ij}	β_{ij}	(s_{ij}, e_{ij})	NA	NA
$\mathbf{P}_i$	Set of α prefix flows	d_i	$\mathbf{p}_{il}$	ζ_{il}	η_{il}	NA	μ_{il}	g_{il}

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}, $v\epsilon(1, u_{il})$. The α-time, μ_{il} (see Table 2), is then computed as

$$\mu_{il} \triangleq \sum_{v=1}^{u_{il}} (e^l_{iv} - s^e_{iv}) + \sum_{u=1}^{y_{il}} (e_{iu} - s_{iu}), 1 \leq l \leq d_i \tag{3}$$

The measures, α-bytes and α-time, are distinct from size and duration. First, α-bytes and α-time are used to characterize α prefix flows, while in another paper [60], we characterized the size and duration of α (5-tuple) flows. Our definitions of the terms "prefix flow" and "5-tuple flow" are provided in Sect. 3.1. Second, the α-bytes measure includes only those bytes sent in intervals in which the H threshold was crossed, while the size of an α flow additionally includes bytes reported by NetFlow records in which the H threshold was not crossed. Similarly, α-time is the cumulative sum of time differences (between the last packet timestamp and the first packet timestamp) from only those NetFlow records in which the H threshold was crossed, while duration of an α flow is the difference between the last packet timestamp of the last NetFlow record and the first packet timestamp of the first NetFlow record of the flow. Thus, duration includes all NetFlow records, not just those in which the H threshold was crossed, and it includes gaps between the last packet timestamp and first packet stamp of consecutive NetFlow records.

6 NetFlow Data Analysis

ESnet NetFlow data was collected from an ESnet provider edge router for seven months: May 1–Nov. 30, 2011 (214 days). In ESnet routers, NetFlow was configured to sample 1-in-1,000 packets, and the active timeout interval was set to 60 seconds. Flow information was collected for the incoming side of all inter-domain interfaces. Flow tools [61], and custom Perl and R [62] programs are used to analyze the data. These programs have been posted as open-source software on our project Web site [14] to enable replication of our tests using NetFlow data from other RENs.

The following parameter values were used: τ, NetFlow active timeout interval, was 1 minute, aggregation interval for creating prefix flows was 1 day, and H, the α NetFlow record threshold, was 1 GB.

We explain our reasons for choosing the 1 GB threshold. Whether an α flow causes increased packet delay/jitter for real-time flows depends upon background traffic, link rates, and router buffer size. Experimental work in another paper [8] showed how a data-transfer node with a 10 Gbps network interface card (NIC) could send back-to-back packets in high-rate large bursts filling router buffers when TCP's congestion window is large. Scientists who repeatedly move large datasets invest in data-transfer nodes with such high-speed NICs. The 1 GB threshold was selected after determining from our experiments [8] that the router buffer size was 125 MB, and using our knowledge of background traffic in ESnet4. If the TCP congestion window size increases to hundreds of MB, packets could arrive in bursts at router interfaces at close to 10 Gbps speeds. In the presence of background traffic, if the egress link rate at the router is also 10 Gbps, router buffers can fill up. Thus large file transfers have the potential to cause increased packet delay/jitter for other flows, and are hence flagged as α flows. For the link rates, NIC rates, and background traffic observed in ESnet4, we selected 1 GB as the threshold.

ESnet recently upgraded its backbone links to 100 Gbps speeds. In a wide-area network test, 94 Gbps end-to-end TCP throughput was demonstrated between clusters located at the Bay Area, CA, and at Chicago, IL [63]. Each cluster consisted of three servers, and each server was equipped with four 10 Gbps NICs. The routers at the two locations were interconnected by a 100 Gbps wide-area circuit. This experiment shows the feasibility of single α flows generating data at rates close to the 100 Gbps rate. As 100 Gbps NICs appear in the market and get deployed, the scenario described above of router buffers filling up will occur again. NIC speeds have traditionally lagged router interface speeds, but typically catch up soon because they are required and used in the scientific community for high-speed file transfers. When NIC speeds match backbone link speeds, there is a need for α-flow traffic engineering.

A new symbol I is used to represent the period of the analyzed NetFlow data, which was 214 days. Two types of prefix identifiers were used: (1) /32 source and destination IP addresses, and (2) /24 source and destination subnet IDs. The aging parameter used to delete entries from the firewall filter was varied to study its impact.

First, in Sect. 6.1, observed α flows are characterized in different ways. Examples of questions answered are what is the maximum per-day total amount of bytes sent in α intervals by any single source–destination host/subnet pair, do source–destination pairs that generate α flows do so repeatedly (on different days), what is the cumulative per-day total amount of bytes across all source–destination pairs that generate α flows, what is the total amount of time where there are one or more active α flows on a router interface, and what percentage of the total traffic is represented by α flows. *Second*, Sect. 6.2 evaluates an effectiveness measure, the percentage of α-bytes that would have been redirected to traffic-engineered paths and isolated from other flows, had AFTES been operational during the May–Nov. 2011 period. While the effectiveness numbers are higher when /24 prefix IDs are

used in firewall filters for α flow redirection and isolation than when /32 prefix IDs are used (91 vs. 82 %), there is a cost to using /24 prefix IDs in that β-flow packets that share the same prefix IDs as α flows will get directed to the traffic-engineered paths/queues set up for α flows. In a *third* set of analysis, Sect. 6.3 quantifies this cost, using a measure of the percentage of these packets that are attributable to file transfer applications (the assumption being that α-flow bursts are less adverse to file transfer flows than to real-time delay-sensitive flows). *Finally*, a hypothesis that α flows primarily originate from "data doors," well-equipped nodes dedicated for large high-speed data transfers, is tested and found to be true.

6.1 Characteristics of Observed α Prefix Flows

As explained in Sect. 5, sets $\mathbf{R}_i$, $1 \le i \le I$, were created from the daily set of NetFlow reports using the α-flow criterion. From this filtered set of α NetFlow reports, per-day α prefix flow sets $\mathbf{P}_i$, $1 \le i \le I$, were created. The following characteristics of these sets are described:

- Data for individual α prefix flows
- A measure of persistence
- Cumulative per-day data
- Relative values of α flows when compared to all traffic

6.1.1 Individual α Prefix Flows

A total of 125 unique /24 α prefix IDs and 1,548 unique /32 α prefix IDs were observed in the 214-day NetFlow reports. In other words, 125 source–destination subnets and 1548 source–destination hosts generated α flows in the observed 214-day period. Not all α prefix IDs made an appearance every day. A matrix consisting of α prefix IDs as rows and the 214 days as columns was sparse, where each matrix entry consists of two tuples: {α-bytes, α-time}.

The data (α-bytes and α-time) corresponding to α prefix flows $\mathbf{p}_{il}$, $1 \le l \le d_i$ and $1 \le i \le I$, for 6 different percentile values are presented in Table 3. For example, $\max_{1 \le i \le I}(\max_{1 \le l \le d_i} \eta_{il})$ for the /24 α prefix IDs is 2.6 TB. In the other days,

Table 3 Data for individual α prefix flows (the per-day sets for the whole 214-day period are sorted by α-bytes or α-time)

	Percentiles	100 %	90 %	80 %	70 %	60 %	50 %
Sorted on α-bytes (GB)	/24	2,603.58	44.43	16.94	8.5	5.34	3.46
	/32	2,060.29	12.54	5.48	3.5	2.44	1.95
Sorted on α-time (min)	/24	544.64	21.27	8.16	4.54	2.66	1.88
	/32	544.64	6.16	2.76	1.78	1	0.99

within that 214-day period, α flows corresponding to one α prefix ID transferred 2.6 TB within α-intervals in one day. The longest α-time from among all prefix flows (/24 or /32) observed in the 214-day period was 544.64 min (i.e., 9.08 hours of α intervals within one 24-h period). This is significant because during α intervals, packets from real-time audio–video flows could have suffered increased delays. These 100 percentile values are not typical. As seen in Table 3, 90 % of α prefix flows have a much smaller α-time, i.e., 21.27 mins for /24 sets and 6.16 mins for /32 sets.

6.1.2 A Measure of Persistence

The number of days in which an α prefix ID makes an appearance is characterized as a measure of persistence. Let **U** represent the set of unique α prefix identifiers that appeared in the 214-day observation period (determined from the sets $\{\zeta_{il}, 1 \le l \le d_i$ and $1 \le i \le I\}$). For each $u \in \mathbf{U}$, N_u is defined as the number of days in which $\{\eta_{il} > 0$ for $\zeta_{il} = u, 1 \le l \le d_i$ and $1 \le i \le I\}$.

Cumulative probability plots of N_u for the /24 and /32 cases are shown in Fig. 4. The maximum number of days (out of 214) in which a /24 α prefix ID appeared was 114. The maximum number of this persistency measure for a /32 α prefix ID was 68. Of the /24 and /32 unique α prefix IDs, 21.6 and 4.5 % appeared more than 15 days, respectively, and 39.2 and 10.34 % appeared more than 7 days, respectively. These numbers show that some source–destination host/subnet pairs repeatedly generate α flows. This is consistent with our hypothesis.

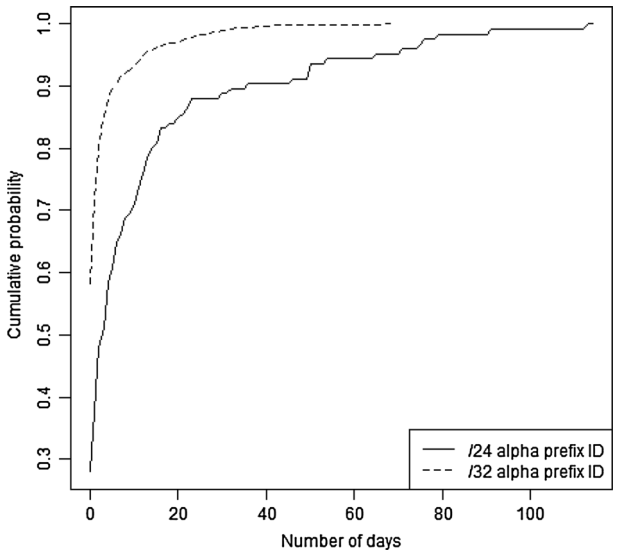

Fig. 4 Cumulative probability of the number of days, $N_u, u \in \mathbf{U}$, in which a unique α prefix ID u made an appearance

6.1.3 Cumulative Per-Day Data

The total α-bytes ($\sum_{l=0}^{l=d_i} \eta_{il}$), total α-time ($\sum_{l=0}^{l=d_i} \mu_{il}$), and total number of unique 5-tuple IDs among the α flows, for each day i, for $1 \leq i \leq I$, are plotted in Fig. 5. Both the total α-bytes and total α-time peaked on Jul. 28, 2011. On this day, there were 2.65 TB transferred in α-intervals, and 9.28 hours of α-time. It was noted earlier (Table 3) that most α prefix flows have small α-times, i.e., 90 % of the /24 sets and /32 sets have per-day numbers less than 21.27 mins and 6.16 mins, respectively. However, the cumulative data analysis illustrates that in 10 % of the days, the total amount of α-time (i.e., the sum of all unique α-intervals) was more than 4.1 hours, which means during these intervals, α flows could cause increases in the delays experienced by real-time audio-video flow packets (this particular router has only one outgoing link to another ESnet router on which all these α flows are being carried). The total number of unique 5-tuple IDs among all α flows peaked to a value of 1662 on Nov. 22, 2011.

In the plots of Fig. 5, one can observe increasing trends in α-flow activity over the 7-month period. For example, it appears that there were more days with α flows in the later months of the observed time period than in earlier months. In order to study this trend for the three measures plotted in Fig. 5, a coarse binary quantization was applied before fitting the points with a smoothing spline function. Specifically, the median value for each of the three measures was chosen as the threshold for a

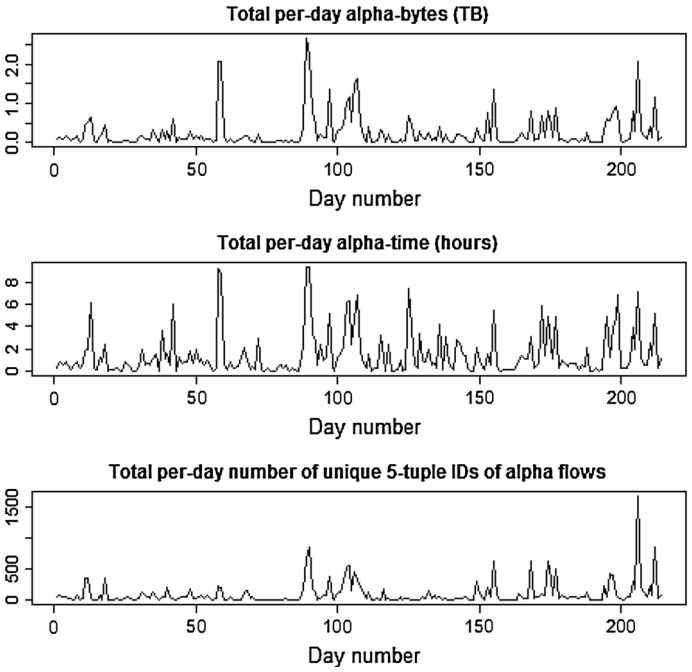

Fig. 5 Cumulative (across all α prefix IDs) per-day data

0/1 quantization of the observed values, and two different values were used for the number of degrees of freedom (*df*), 2 and 4, in the smoothing spline function. Lower values of *df* creates greater smoothing at a cost of accuracy of fit. The increasing trends seen in the smoothed spline plots of Figs. 6 and 7 are likely due to increasing sizes of scientific data sets, and frequency of transfers.

6.1.4 Relative Values of α Flows When Compared to All Traffic

Table 4 shows the portions of the total traffic constituted by α-bytes on a monthly basis for the observed time period. The total traffic was determined from SNMP link usage data collected by ESnet [18]. The percentages for all seven months were less than 2 %. Prior work cite the 50–20 rule [64], in which 20 % of the flows contribute 50 % of the packets. As noted in Sect. 3.2, the criterion used for classifying a flow report as belonging to an α flow did not rely on a volume threshold such as the 50 % number used by Lan and Heidemann [30]. Instead the threshold was fixed at 1 GB in a time period less than equal to 1 min. This threshold was high enough that the byte ratio is small at 2 %. An analysis of actual GridFTP usage transfer statistics for two source–destination pairs [2] shows that the percentage of scientific transfers that would create flows exceeding this threshold is small. This high threshold was selected because of the light link loads observed with SNMP data, which indicates that unless a sizeable burst occurs in a short duration the flow is not likely to cause adverse effects on real-time flows, and hence should not be labeled an α flow. Even

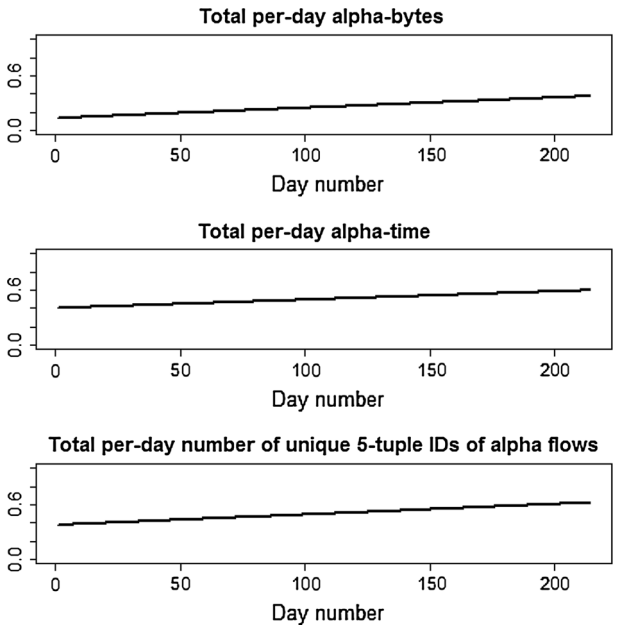

Fig. 6 Binary quantized cumulative data plus a smoothing spline function with $df = 2$

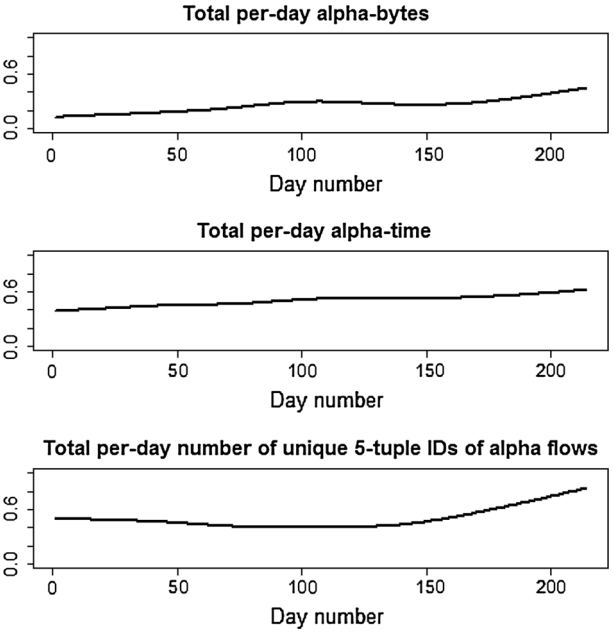

Fig. 7 Binary quantized cumulative data plus a smoothing spline function with $df = 4$

Table 4 Percentage of α-bytes in total traffic for /24 case

Month (2011)	May	Jun	Jul	Aug	Sep	Oct	Nov
α-bytes (TB)	4.2	7.8	7.9	10.9	5.2	6.9	10.3
Total (TB)	625.9	533.7	692.3	640.6	740.6	1,101.8	869.9
Percentage	0.67	1.46	1.14	1.7	0.7	0.63	1.18

though the percentage of α bytes was small, the high-rate/large size of α flows can have adverse effects as illustrated in Sect. 2.

6.2 Effectiveness of AFTES

This analysis measures the effectiveness of the offline approach proposed for AFTES. *Effectiveness* is defined as the percentage of α-bytes that would have been redirected to traffic-engineered paths and isolated from the general traffic had AFTES been deployed. But before presenting these percentages, the numbers of **new** α prefix IDs that appeared each day are presented as a test of the hypothesis that α flows are created repeatedly between the same source–destination pairs. After presenting the effectiveness values, an analysis of the impact of the aging parameter is presented.

6.2.1 Number of New α Prefix IDs Observed Each Day

Figure 8 shows the number of new α-flow associated prefix identifiers appearing each day in the 214-day observed period. The aging parameter A was set to 214 days (i.e., firewall filter rules corresponding to α flows were never deleted). The numbers would be greater if the aging parameter was set to a smaller time period to prevent the firewall filter from growing too large. The dependence on the aging parameter is presented later. Overall, the trend in the new α prefix IDs graph is downward as seen with the smoothing spline function (degrees of freedom set to 4) in Fig. 9. For example, on day 1, there were 9 new /24 α prefix IDs but after day 45, in 94.7 % of the days, there were only 0 or 1 new α prefix IDs. The implication is that the size of the firewall filter needed to support traffic engineering for α flows may not be significant, since modern routers allow for very large numbers of firewall filter rules. However, there can be days, such as Nov. 10th, 2011, when α flows were observed corresponding to 6 new /24 prefix IDs, and 141 new /32 prefix IDs. This can happen when there are new installations of high-speed data transfer nodes or when new scientists access existing data transfer nodes.

6.2.2 Effectiveness

Figure 10 shows the effectiveness of AFTES; specifically it shows the percentage of α-bytes that would have been redirected and isolated, on a per-month basis. Table 5 shows the aggregate percentage of α-bytes that would have been directed across the whole observed time period of 214 days corresponding to different values of the aging parameter, A. Usage of /24 prefix IDs was more effective than the /32 prefix IDs (the negative aspect of this finding is quantified in the next section). This is to be expected since data transfer nodes are often cluster computers with IP addresses in

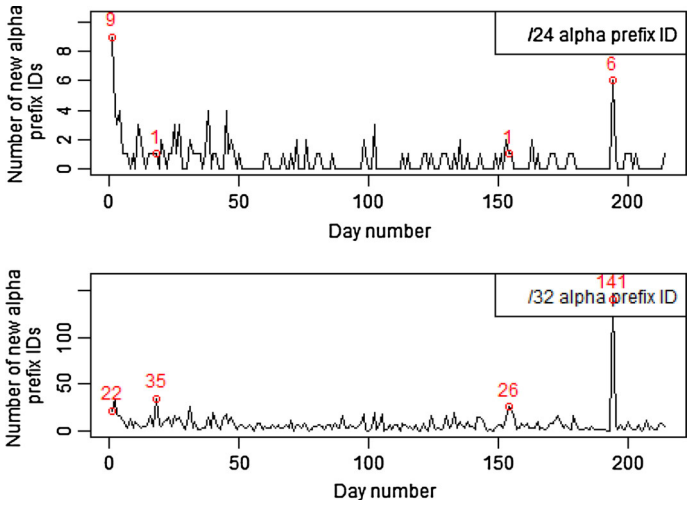

Fig. 8 Number of new α prefix IDs per day

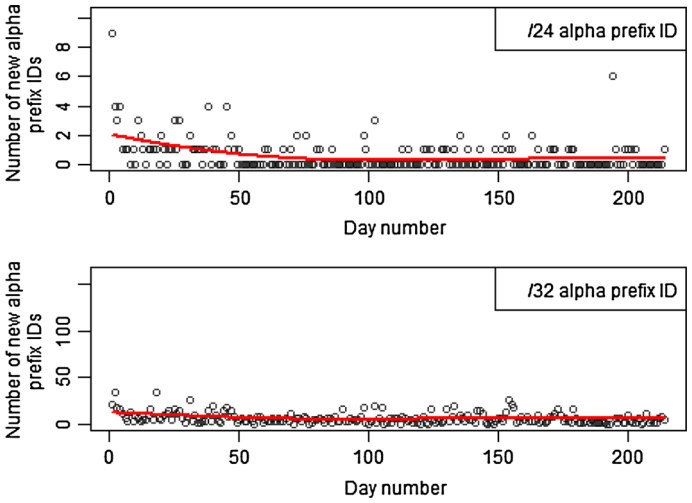

Fig. 9 Number of new α prefix IDs per day with smoothing spline function ($df = 4$)

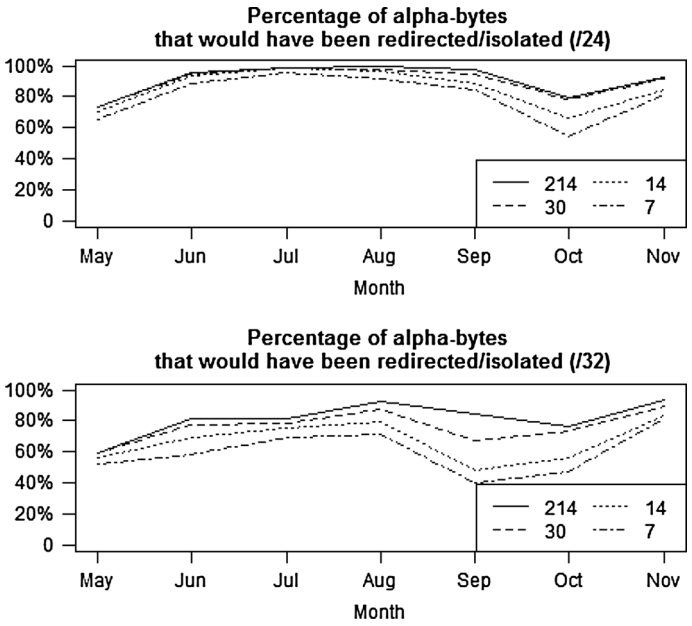

Fig. 10 Effectiveness of AFTES for different values of the aging parameter

the same subnet. With new installations of high-speed data transfer nodes, previously unseen /32 prefix identifiers would have been covered by /24 identifiers. If rules are never deleted from the firewall filter, for the /24 case, 92 % of α-bytes would have been isolated from β flows across the 7 months.

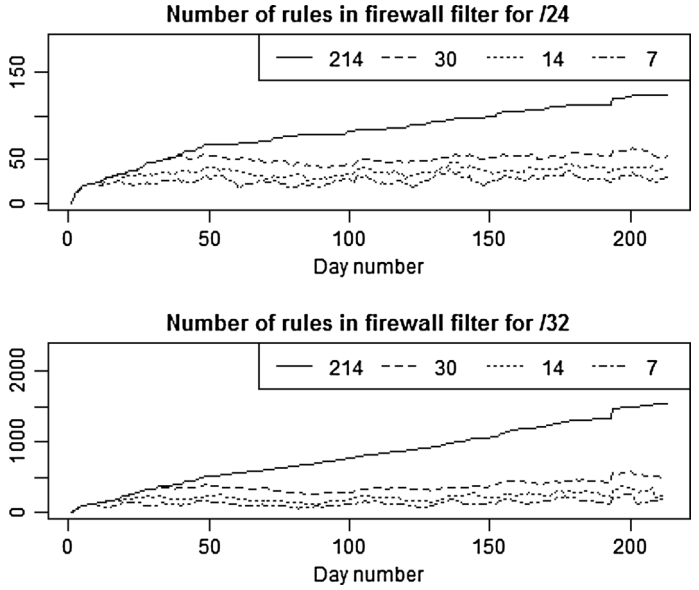

Fig. 11 Growth of firewall filter for different values of the aging parameter [6]

Table 5 Percentage of α-bytes that would have been redirected and isolated across whole 214-day period

Aging parameter (days)	/24 (%)	/32 (%)
7	82	67
14	87	73
30	91	82
214	92	86

6.2.3 Impact of aging parameter

The negative of never deleting firewall filter rules is that the firewall filter keeps growing as shown in Fig. 11, though the absolute size was small relative to what modern routers can support. But for sustainability, firewall filter rules corresponding to prefix IDs with no α flow appearances should be deleted. The use of an aging parameter of 30 days appears to be a good compromise. The firewall filter size levels off as seen in Fig. 11, while at the same time, the effectiveness measure does not decrease significantly. The aggregate percentage across 7 months decreases from 92 to 91 % for the /24 case, when the aging parameter was dropped from 214 to 30 days, as seen in Table 5.

6.3 Quantifying the cost of AFTES on β Flows

The cost of using /24 source and destination subnet identifiers rather than the more constrained /32 host identifiers in firewall filters is that β flows that share the same

source/destination address prefixes as α flows will also get redirected to the traffic-engineered paths/queues meant to handle α flows. Packets from these β flows could suffer increased delays as they would be queued in the same buffers as those used for α flows (see Sect. 2). The purpose of this analysis was to characterize these unfortunate β flows. The term "hapless β flows" is used for these flows that share α prefix identifiers, and get traffic engineered on to the same paths and queues as α flows.

The method used was to start with the set of non-α NetFlow records (flow records in which the size threshold H is not exceeded) for each aggregation interval i, and then apply three filters in sequence. First, non-α NetFlow records corresponding to α flows were identified. As noted in Sect. 5 and per the definitions in Table 1, a 5-tuple flow is classified as an α flow even if it has just one α NetFlow record. Other NetFlow records corresponding to this 5-tuple flow may have sizes that are below the H threshold. Therefore, the first step was to identify the subset of non-α NetFlow records that belong to α flows. The remaining non-α NetFlow records were from hapless β flows.

The second filter was used to identify NetFlow records from file transfer applications with the assumption that the delay impact on such flows caused by bursts from α flows is not as critical as on real-time flows. The criteria used for a NetFlow record to qualify as being from a file transfer application were that (a) the number of bytes/packet should be at least 1000, (b) the total byte count should be greater than a threshold G, where $G < H$, and (c) there is at least one other NetFlow record within the aggregation interval that shares the same source and destination IP addresses, protocol type, and at least one of the two port numbers as the candidate NetFlow record. Typically NetFlow records from file transfer applications will have multiple packets in spite of the low sampling rate (e.g., 1-in-1,000), and most of these packets will be maximum-sized packets (Ethernet's Maximum Transmission Unit size is 1,500 B). The second criterion ensures that the flow corresponding to the flow record was generating a sizeable amount of data. The last criterion was applied because parallel TCP streams are used by scientific data transfer applications such as GridFTP. For example, 8-stream transfers were common [2]. Flows corresponding to these 8 streams typically shared the source and destination IP addresses, protocol type (TCP), and either the source port number or destination port number, with the other port number being different for each of the 8 streams. Therefore, due to the low sampling rate, even if there was just one NetFlow record from one stream, it is likely that there were NetFlow records from the other streams.

The third filter isolated NetFlow records for flows with well-known port numbers; specifically ssh, http, imap, smtp, ssmtp, http, https, nntp, imap, imaps, imap4ssl, unidata, rtsp, rsync, sftp, bftp, ftps, pop3, and sslpop. Some file transfers used scp, which would appear as ssh flows, but such flows would have been filtered out into the first or second categories leaving behind only those ssh flows corresponding to interactive applications such as SecureCRT because of the sequential application of these filters.

After the sequential application of the three filters on the set of non-α NetFlow records for each aggregation interval i, the remaining (unclassified) NetFlow records were grouped together as "Leftover." The set of non-α NetFlow records

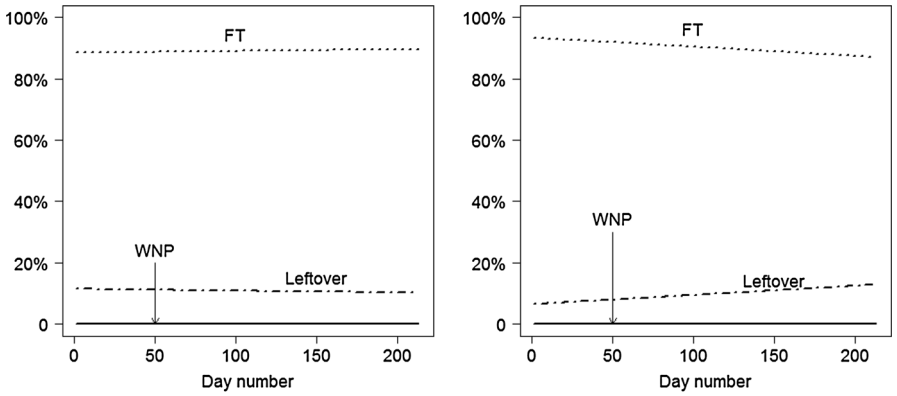

Fig. 12 Percentages of four groups of hapless β flow packets with smoothing spline function ($df = 2$). **a** /24 prefix ID case, **b** /32 prefix ID case

corresponding to hapless β flows were thus divided into three groups: File Transfers (FT), Well-Known Ports (WNP), and Leftover.

Our analysis focused on packets instead of bytes for these three groups of hapless β-flow Netflow records. This is because for smaller sized flows, the size accuracy ratio of multiplying the byte count by 1000 (because of the 1-in-1,000 packet sampling) will be lower than for large-sized flows [7]. Specifically, for the sampled set of packets from the hapless β flows, the percentages represented by packets from each of the three groups were computed. Across the whole 214-day period, most of the hapless β flow packets belonged to the first group (file transfers). For the /24 case, 89.37 % of the hapless β-flow packets were classified as being from file transfer flows, and for the /32 case, the percentage was 88.77 % with the G threshold set to 10 MB (recall the H threshold was 1 GB). Only a small percentage of the hapless β-flow packets belonged to the second group, i.e., from non-file transfer flows with well-known port numbers: for the /24 this number was 0.026 % for both the /24 and /32 cases. The third group (leftover) percentage was small (e.g., 10.6 % for the /24 case).

The per-day percentages for these three groups of hapless β flow packets are plotted using smoothing spline functions (degrees of freedom set to 2 and 4), in Figs. 12 and 13. For both /24 and /32 cases, the WNP group percentages were almost 0.

In summary, the above analysis shows that most β flows that shared α prefix identifiers were from file transfer applications, and not from interactive applications such as VoIP and Web browsing. This is likely because in most supercomputing centers, high-end data transfer nodes are set up as clusters in their own subnets. Therefore the cost of using prefix IDs instead of 5-tuple flow IDs for traffic engineering α flows appears to be low.

Furthermore, we recommend the use of /24 address prefixes rather than /32 addresses because the effectiveness measure was higher for the former, and the

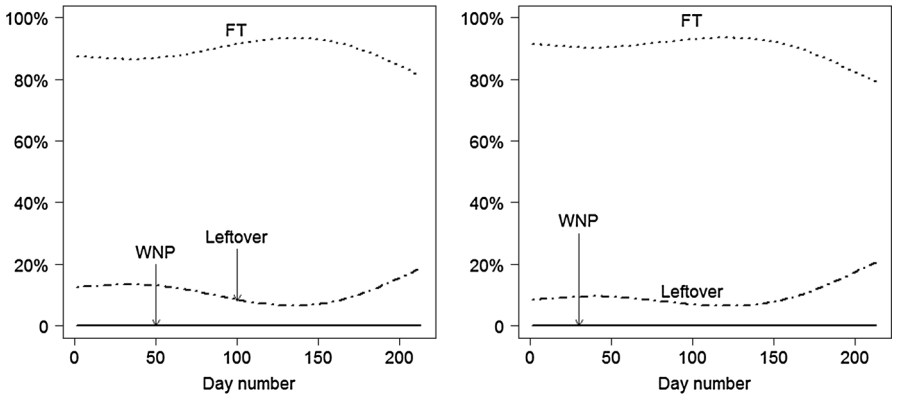

Fig. 13 Percentages of four groups of hapless β flow packets with smoothing spline function ($df = 4$). **a** /24 prefix ID case, **b** /32 prefix ID case

negative effect on β flows was small. It is better to have a small percentage of β flows be impacted by α flows (which will happen at a higher rate with /24 prefixes) than to miss α flows, which will happen at a higher rate with /32 addresses, and have those α flows impact the much larger proportion of β flows on the default IP-routed paths.

6.4 Data-Door Analysis

We hypothesize that α flows are created by large file transfers from well-equipped nodes that are referred to as "data doors." In order to test this hypothesis, the domain names/IP addresses of data doors in major high performance computing facilities (e.g., rftpexp.rhic.bnl.gov) were obtained from their corresponding Web sites, and these IP addresses were matched against the α prefix identifiers found through the above described NetFlow data analysis. Our findings support the hypothesis.

Absolute and relative measures of the daily number of α prefix IDs matched with data door address prefixes are shown in Fig. 14. In 96 (44.9 %) days out of 214, all α prefix identifiers matched with those of data doors, and in 175 (81.8 %) days out of the 214 days, the matched rate was more than 80 %.

Absolute and relative measures of the per-day α-bytes from/to data doors are shown in Fig. 15. In 148 (69.2 %) days out of 214, more than 90 % of the α-bytes were from/to data doors.

Histograms of the two ratios presented above are shown in Fig. 16. The high probability mass at a ratio of 1 indicates that for most days, the α flows came from data doors. In summary, it appears that indeed most α prefix flows were from/to high-end servers that are designed specifically for handling data transfers.

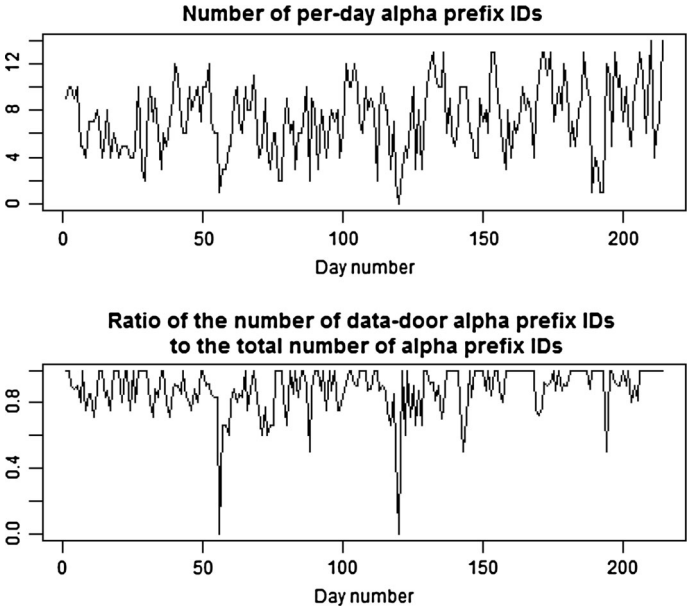

Fig. 14 A measure of the ratio of α prefix IDs from/to data doors

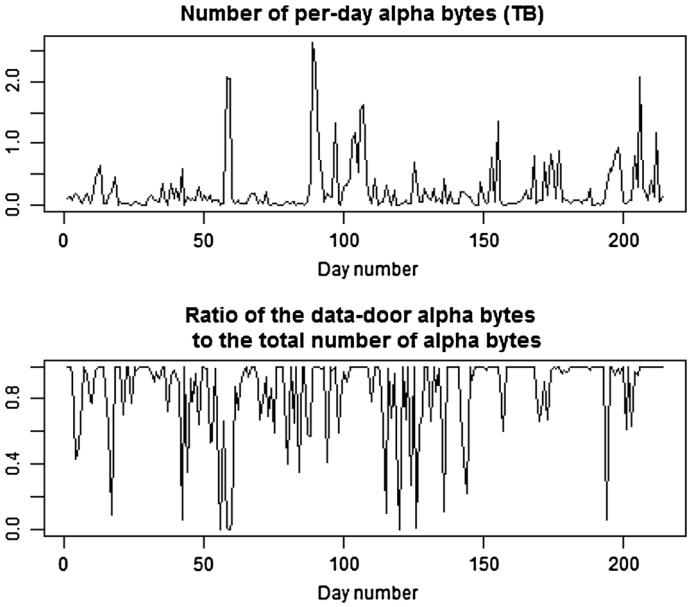

Fig. 15 A measure of the ratio of α-bytes from/to data doors

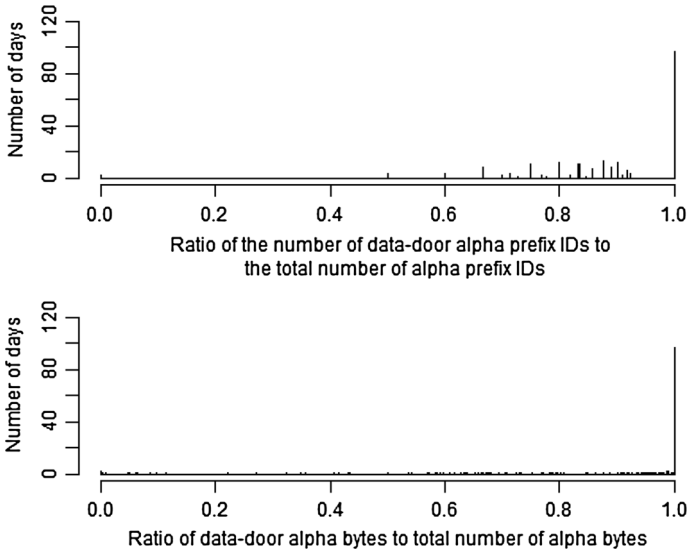

Fig. 16 Histograms of the two data-door ratio measures

7 Summary and Conclusions

An offline mechanism was developed for determining prefix identifiers of α flows, which are caused by high-rate, large-sized data transfers that are typically created in research-and-education networks by scientific researchers. Such an offline scheme is developed for a network management system called AFTES for intra-domain traffic engineering. Prefix identifiers (IDs) (source and destination address prefixes) of persistent α flows are used to set firewall filters to redirect future α flows to traffic-engineered paths, and to isolate their packets to separate queues. These actions would reduce adverse effects that α flows may have on packets from real-time flows, such as increased packet delay and delay variance. The effectiveness and cost to β flows of this offline mechanism was evaluated through an analysis of 7 months of NetFlow data obtained from an ESnet router. Our conclusions are (1) scientists who move large datasets at high rates do so repeatedly between the same source-destination pairs, and (2) subnets with high-end data-transfer nodes are typically distinct from user desktop subnets, which limits the percentage of real-time β-flow packets that share α-flow address prefixes. These observations support the viability of the AFTES solution for provider deployment.

8 Future Work

Future work will address the question of how AFTES can be integrated into OpenFlow/SDN networks. In current deployed OpenFlow/SDN networks, if NetFlow is supported by the switches/routers, then our offline AFTES can be directly tested. If

the hypothesis stated in Sect. 1, with regards to the traffic patterns of α flows, holds in the tested OpenFlow/SDN network, then our offline AFTES can be deployed.

Further, it may be possible to design an online AFTES for OpenFlow/SDN networks as the time needed to program switches for redirection of an α flow by an SDN-controller could be shorter than in today's routers, where the "commit" operation used to modify router configurations may require a few seconds. As noted in Sect. 4, in our companion paper [8], we described our approach to router configuration for handling α flows, which consist of establishing MPLS LSPs and configuring firewall filters using JunOS commands (Juniper routers were deployed in ESnet4). A commit operation was required in the JunOS version used in our experiments. A subject for further study is to determine how best to combine offline and online methods to manage and limit the adverse effects of high-rate large-sized flows.

Finally, better packet sampling algorithms than are available in the built-in NetFlow implementations of existing routers could be implemented in OpenFlow/SDN networks by leveraging Network Functions Virtualization (NFV). For example, flow-specific sampling could be implemented to limit packet sampling to flows between potential α-flow source–destination pairs.

References

1. USDOE Office of Science ASCR: Terabit Networks for Extreme-Scale Science Workshop Report (2011). http://science.energy.gov/~/media/ascr/pdf/program-documents/docs/Terabit_networks_workshop_report.pdf
2. Liu, Z., Veeraraghavan, M., Yan, Z., Tracy, C., Tie, J., Foster, I., Dennis, J., Hick, J., Li, Y., Yang, W.: On using virtual circuits for GridFTP transfers. In: The International Conference for High Performance Computing, Networking, Storage and Analysis 2012 (SC 2012), pp. 81:1–81:11, Nov 10–16, 2012
3. Leith, D., Shorten, R.: H-TCP: TCP for high-speed and long-distance networks. In: Protocols for Fast Long Distance Networks Workshop (PFLDnet), Feb 16–17, 2004
4. Sarvotham, S., Riedi, R., Baraniuk, R.: Connection-level analysis and modeling of nework traffic. In: ACM SIGCOMM Internet Measurement Workshop 2001, pp. 99–104, Nov 2001
5. ESnet. http://www.es.net/
6. Jin, T., Tracy, C., Veeraraghavan, M., Yan, Z.: Traffic engineering of high-rate large-sized flows. In: 2013 IEEE 14th International Conference on High Performance Switching and Routing (HPSR), pp. 128–135 (2013)
7. Yan, Z., Tracy, C., Veeraraghavan, M.: A hybrid network traffic engineering system, In: Proceedings of the IEEE 13th High Performance Switching and Routing (HPSR), Jun 24–27, 2012
8. Yan, Z., Veeraraghavan, M., Tracy, C., Guok, C.: On how to provision Quality of Service (QoS) for large dataset transfers, In: Proceedings of the Sixth International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ), Apr 21–26, 2013
9. GridFTP. http://globus.org/toolkit/docs/3.2/gridftp/
10. The Lambda Station Project. http://www.lambdastation.org/
11. TeraPaths: Configuring End-to-End Virtual Network Paths with QoS Guarantees. https://www.racf.bnl.gov/terapaths/
12. Circuit Switched High-speed End-to-End Transport Architecture (CHEETAH). http://www.ece.virginia.edu/cheetah/
13. NetFlow. http://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html
14. Hybrid Network Traffic Engineering System (HNTES). http://www.ece.virginia.edu/mv/research/DOE09/index.html

15. Spragins, J.: Asynchronous transfer mode: solution for broadband ISDN, third edition [New Books]. IEEE Netw. **10**, 7 (1996)
16. Braden, R., Zhang, L., Berson, S., Herzog, S., Jamin, S.: Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification. RFC 2205 (Proposed Standard), Sept 1997. Updated by RFCs 2750, 3936, 4495, 5946, 6437
17. Vietzke, R.P.: Internet2 Headroom Practice, 15 Aug 2008. https://wiki.internet2.edu/confluence/download/attachments/17383/Internet2+Headroom+Practice+8-14-08.pdf?version=1
18. ESnet Graphite. https://stats.es.net/graphite/
19. nuttcp. http://www.nuttcp.net/
20. Wallerich, J., Dreger, H., Feldmann, A., Krishnamurthy, B., Willinger, W.: A methodology for studying persistency aspects of Internet flows. ACM SIGCOMM Commun. Rev. **35**(2), 23–36 (2005)
21. ESnet Backbone Topology Map Summer 2010. http://es.net/introducing-esnet5/network-maps/historical-network-maps/
22. GEANT2. http://www.geant2.net/
23. Interoperable Ondemand Network (ION). http://www.internet2.edu/products-services/advanced-networking/layer-2-services/
24. Next-generation network testbed JGN-X. http://www.jgn.nict.go.jp/english/
25. On-Demand Secure Circuits and Advance Reservation System (OSCARS). http://www.es.net/OSCARS/docs/index.html
26. Liakopoulos, A., Maglaris, B., Bouras, C., Sevasti, A.: Providing and verifying advanced IP services in hierarchical DiffServ networks-the case of GEANT. Int. J. Commun. Syst. **17**(4), 321–336 (2004)
27. Claise, B.: Cisco Systems NetFlow Services Export Version 9. RFC 3954 (Informational), Oct 2004
28. Claise, B.: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information. RFC 5101 (Proposed Standard), Jan 2008
29. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An Architecture for Differentiated Service. RFC 2475 (Informational), Dec 1998. Updated by RFC 3260
30. Lan, Kun-chan, Heidemann, John: A measurement study of correlations of Internet flow characteristics. Comput. Netw. **50**(1), 46–62 (2006)
31. Crovella, M.E., Taqqu, M.S.: Estimating the heavy tail index from scaling properties. Methodol. Comput. Appl. Probab. **1**, 55–79 (1999)
32. Brownlee, N., Claffy, K.: Understanding Internet traffic streams: dragonflies and tortoises. IEEE Commun. Mag. **40**, 110–117 (2002)
33. Nguyen, T.T.T., Armitage, G.J.: A survey of techniques for Internet traffic classification using machine learning. IEEE Commun. Surv. Tutor. **10**(4), 56–76 (2008)
34. Awduche, D.O., Jabbari, B.: Internet traffic engineering using multi-protocol label switching (MPLS). Comput. Netw. **40**(1), 111–129 (2002)
35. Wang, N., Ho, K., Pavlou, G., Howarth, M.: An overview of routing optimization for Internet traffic engineering. IEEE Commun. Surv. Tutor. **10**(1), 36–56 (2008)
36. Papagiannaki, K., Taft, N., Bhattacharyya, S., Thiran, P., Salamatian, K., Diot, C.: A pragmatic definition of elephants in Internet backbone traffic, In: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment (IMW '02), pp. 175–176 (2002)
37. Callado, A., Kamienski, C., Szabo, G., Gero, B., Kelner, J., Fernandes, S., Sadok, D.: A survey on Internet traffic identification. IEEE Commun. Surv. Tutor. **11**, 37–52 (2009)
38. Kamiyama, N., Mori, T.: Simple and accurate identification of high-rate flows by packet sampling, In: Proceedings of INFOCOM 2006. 25th IEEE International Conference on Computer Communications, pp. 1–13 (2006)
39. Mori, T., Uchida, M., Kawahara, R., Pan, J., Goto, S.: Identifying elephant flows through periodically sampled packets, In: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement (IMC '04) (New York, NY, USA), pp. 115–120, ACM (2004)
40. Zhang, Y., Fang, B., Zhang, Y.: Identifying high-rate flows based on bayesian single sampling, In: 2010 2nd International Conference on Computer Engineering and Technology (ICCET), vol. 1, pp. V1-370–V1-374 (2010)
41. Duffield, N., Lund, C., Thorup, M.: Estimating flow distributions from sampled flow statistics. IEEE/ACM Trans. Netw. **13**(5), 933–946 (2005)
42. Fioreze, T., Pras, A.: Self-management of hybrid optical and packet switching networks. In: 2011 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 946–951 (2011)

43. Caria, M., Jukan, A.: A novel approach to accurately compute an IP traffic matrix using optical bypass. In: 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), pp. 1135–1141 (2013)
44. Lu, Y., Wang, M., Prabhakar, B., Bonomi, F.: ElephantTrap: A low cost device for identifying large flows. In: 15th Annual IEEE Symposium on High-Performance Interconnects, 2007 (HOTI 2007), pp. 99–108 (2007)
45. Kodialam, M., Lakshman, T.V., Mohanty, S.: Runs based traffic estimator (rate): a simple, memory efficient scheme for per-flow rate estimation. In: INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1808–1818 (2004)
46. Hao, F., Kodialam, M., Lakshman, T.V., Zhang, H.: Fast, memory-efficient traffic estimation by coincidence counting, In: Proceedings of IEEE INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 2080–2090 (2005)
47. Zadnik, M., Canini, M., Moore, A., Miller, D., Li, W.: Tracking elephant flows in Internet backbone traffic with an FPGA-based cache. In: International Conference on Field Programmable Logic and Applications, 2009 (FPL 2009), pp. 640–644 (2009)
48. Paisley, J., Sventek, J.: Real-time detection of grid bulk transfer traffic, In: 10th IEEE/IFIP Network Operations and Management Symposium (NOMS), pp. 66–72, Apr 2006
49. Hohn, N., Veitch, D.: Inverting sampled traffic. IEEE/ACM Trans. Netw. **14**(1), 68–80 (2006)
50. Chen, K., Singla, A., Singh, A., Ramachandran, K., Xu, L., Zhang, Y., Wen, X., Chen, Y.: Osa: An optical switching architecture for data center networks with unprecedented flexibility, In: Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, pp. 18–18, USENIX Association (2012)
51. Farrington, N., Porter, G., Radhakrishnan, S., Bazzaz, H., Subramanya, V., Fainman, Y., Papen, G., Vahdat, A.: Helios: a hybrid electrical/optical switch architecture for modular data centers, In: ACM SIGCOMM Computer Communication Review, vol. 40, pp. 339–350, ACM (2010)
52. Wang, G., Andersen, D., Kaminsky, M., Papagiannaki, K., Ng, T., Kozuch, M., Ryan, M.: c-through: Part-time optics in data centers. In: ACM SIGCOMM Computer Communication Review, vol. 40, pp. 327–338, ACM (2010)
53. Open Networking Foundation. https://www.opennetworking.org/
54. Software-Defined Networking (SDN). https://www.opennetworking.org/sdn-resources/sdn-definition
55. Qazi, Z.A., Lee, J., Jin, T., Bellala, G., Arndt, M., Noubir, G.: Application-awareness in sdn, In: Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM (SIGCOMM '13) (New York, NY, USA), pp. 487–488, ACM (2013)
56. Wang, G., Ng, T.E., Shaikh, A.: Programming your network at run-time for big data applications, In: Proceedings of the First Workshop on Hot Topics in Software Defined Networks (HotSDN '12) (New York, NY, USA), pp. 103–108, ACM (2012)
57. Xiao, X., Hannan, A., Bailey, B., Ni, L.: Traffic engineering with MPLS in the internet. IEEE Netw. **14**(2), 28–33 (2000)
58. Paolucci, F., Cugini, F., Giorgetti, A., Sambo, N., Castoldi, P.: A survey on the path computation element (pce) architecture. IEEE Commun. Surv. Tutor. **15**, 1819–1841 (2013)
59. Sharma, A., Mishra, A., Kumar, V., Venkataramani, A.: Beyond MLU: An application-centric comparison of traffic engineering schemes. In: 2011 Proceedings of the IEEE INFOCOM, pp. 721–729, IEEE (2011)
60. Jin, T., Tracy, C., Veeraraghavan, M.: Characterization of high-rate large-sized flows. In: 2014 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), pp. 73–76 (2014)
61. Flow-tools. http://www.splintered.net/sw/flow-tools/docs/flow-tools.html
62. The R Project for Statistical Computing. http://www.r-project.org/
63. Balman, M., Pouyoul, E., Yao, Y., Bethel, E.W., Loring, B., Prabhat, M., Shalf, J., Sim, A., Tierney, B.L.: Experiences with 100 gbps network applications, In: Proceedings of the Fifth International Workshop on Data-Intensive Distributed Computing Date (DIDC '12) (New York, NY, USA), pp. 33–42, ACM (2012)
64. Thompson, K., Miller, G., Wilder, R.: Wide-area Internet traffic patterns and characteristics. IEEE Netw. **11**, 10–23 (1997)

Zhenzhen Yan received her B.S. degree in Electrical Engineering from China University of Geosciences in 2006, and M.S. degree in Computer Engineering from Old Dominion University in 2008. She joined

University of Virginia as a Ph.D. student in 2008. Her research interests include traffic anallysis, traffic engineering, and network management.

Chris Tracy is a Network Engineer at the Energy Sciences Network (ESnet) in the Scientific Network Division of the Lawrence Berkeley National Laboratory. He has worked in computing and networking since the mid-90s. Prior to ESnet, he was an engineer at Mid-Atlantic Crossroads (MAX) GigaPOP at the University of Maryland, focusing on the NSF DRAGON project to deploy experimental optical networks for advanced e-science applications. Tracy received his B.Sc. in Computer Engineering from the University of Pittsburgh in 2001, and is currently pursuing a masters degree at University of Maryland University College (UMUC).

Malathi Veeraraghavan is a Professor in the Charles L. Brown Department of Electrical and Computer Engineering at the University of Virginia. After a ten-year career at Bell Laboratories, she served on the faculty at Polytechnic University, Brooklyn, New York from 1999-2002. She served as Director of the Computer Engineering Program at UVa from 2003- 2006. She holds twenty-nine patents, has over 90 publications, and has received six Best-paper awards. Most recently, she served as the Technical Program Committee Co-Chair for the NGN Symposium at IEEE ICC 2013.

Tian Jin received the B.S degree in Computer Science from Nanjing University, China, in 2011. Since 2011, she has been working as a Ph.D student in the Department of Computer Science, University of Virginia. Her research interests are in the field of traffic characterization and traffic engineering.

Zhengyang Liu received his B.Sc. in Computing from The Hong Kong Polytechnic University in 2006. He joined University of Virginia as a Ph.D. student in 2011. His research interests include high-speed networks and traffic characterization.