Network Traffic as a Federated Testbed Service

Jack Brassil
Dept. of Computer Science
Princeton University
Princeton, NJ, USA
jbrassil@princeton.edu

***Abstract*—We describe a prototype of *Science Traffic as a Service* (STAAS), a decentralized, cooperative system to collect, filter and distribute a diverse collection of real and synthetic data traffic to the global experimental research testbed user community. Available on-demand to networking experimenters through a web dashboard, the tool promises to elevate traffic selection and distribution to a first class experimental instrumentation resource. We believe the alternatives to providing this service on large-scale federated testbeds are increasingly unworkable for experimenters. As backbone networks increasingly deploy 100-1000 Gbps communications links we are moving beyond the point where experimenters can reasonably be asked to independently, safely, and efficiently create test traffic that provides the realism that their investigations will demand. We seek to deploy our prototype at campuses and testbeds attached to the emerging *FABRIC* mid-scale networking research infrastructure. We describe prototype design, operation and implementation, and how it is integrated with existing campus networking infrastructure. We explain how remote experimenters will request and acquire network traffic to study. We detail our process for forwarding campus traffic onto the experimental testbed, while striving to preserve both the timing integrity of the flows and the data privacy of their payloads.**[†]

***Index Terms*—experimental testbeds, federation, data traffic-as-a-service, terabit networks, FABRIC**

I. Introduction

Experimental wide-area networking research infrastructures have proven invaluable for advancing networking research and education. But many experimenters report that previously developed testbeds have been hard-to-use. We address this concern by developing and deploying a new, scalable instrumentation tool to support experimentation on current and future testbeds. We are constructing a realistic, high-performance, system-wide traffic generation and distribution service to be deployed on campus edge nodes throughout the *FABRIC* [1] infrastructure, a shared, national-scale programmable networking research infrastructure (Fig. 1).

FABRIC is a federated testbed-of-testbeds. It enables an experimenter to construct isolated network topologies. A variety of virtualization technologies (e.g., VLANs, physical channels, time-/space-/frequency-multiplexing) support the separation of running experiments in what is viewed by each experimenter as his or her own experiment *container* or *slice*. Though connected to University campuses, other research testbeds, the global public internet, and public cloud cloud providers, a container carries no production data traffic. But testing advances in networking technologies and protocols in the presence of other data traffic is vital to experimenters. Hence, testbed experimenters seek a high degree of control of the exposure of their admitted traffic to interfering cross-traffic, ideally seeking cross-traffic representative of their target operating network environment.

Exposure of an experiment to external traffic offers realism, where the experimenter potentially beneficially learns unanticipated affects of cross-traffic on foreground traffic. But experiment control, reproducibility and understanding are facilitated in the absence of uncontrolled external traffic (i.e., idealism). Both environments are valuable to experimenters. Earlier networking research infrastructures have chosen to follow either approach. For example, PlanetLab [2] operated entirely "in the wild" on the commercial internet, embracing the mixing of experimental traffic with other campus-based and wide-area traffic. GENI [3] pursued a similar course by connecting sites over Internet2's Advanced Layer 2 Service [4]. In contrast, Emulab [5] kept emulated WAN technologies free of traffic to enable complete experimenter control.

In such an isolated container an experimenter's own injected background traffic [6] permits some degree of 'controlled realism.' But numerous investigations have shown the challenges of generating up-to-date, realistic synthetic workloads at scale [7]. Whether artificial traffic generation at scale can accurately model the 'essential characteristics' of some targeted real traffic, and how this can be cheaply and easily

Fig. 1: *FABRIC* Phase 1 nationwide network topology.

[†]This material is based upon work supported by the National Science Foundation under Grant No. CNS-2018308.

realized, remains an open question. This challenge can be eased somewhat by generating only application-specific background traffic [8], though this approach might detract from the representativeness of background traffic observed in most settings. Replay of previously captured live network packet traces represents another approach to synthesizing background traffic. However replaying existing traces raises questions about representativeness and timeliness, as many traces are outdated. Further, the storage resources required to support replay of long duration, high speed traces can be prohibitively expensive.

Many FABRIC experimenters will desire to test their work in the presence of realistic, representative background traffic. Indeed, some networking research such as the identification and investigation of heavy hitter flows [9] *relies* on the presence of external background traffic to study adequately. But as the bandwidth of network links grows into terabits, generating material background traffic becomes more difficult, and in some cases independently infeasible. Consider the plight of a researcher connected via a 10 Gbps campus connection seeking to congest and investigate a 400 Gbps core FABRIC link.

To address this challenge this paper describes a work-in-progress prototype of a cooperative traffic generation system called *STAAS* that raises experimental data traffic distribution to a first-class system-wide service – one that will reduce the experimenter's workload, improve the experimenter's User eXperience (UX), and facilitate experiment reproducibility. Our key insight is that plenty of data flows – often "science" flows – are already in transit at any moment, either on campus or inbound/outbound at the campus border. Rather than synthesize all new traffic, we propose to tap available active flows, say inbound traffic to a campus Science DMZ (SDMZ) carrying a public science data set, and mirror it onto the experimental testbed. Scale is achieved by enlisting participation of many or all cooperating edge nodes in the traffic generation system. To do this safely, our system will permit traffic redistribution only when 1) offered for redistribution by an originating end system (e.g., campus or testbed), and 2) specifically requested by an authorized experimenter. The next section discusses overall system design and capabilities, and Sect. III details its implementation. The planned integration with the FABRIC testbed is discussed in Sect. IV, and the following section discusses the thorny issues of data ownership and privacy, security, and participation incentives.

II. System Design and Operation

A testbed experimenter using STAAS can request several types of offered network traffic; each is appropriate for a specific experiment requirement. Offered traffic might correspond to a single flow, or an aggregate of many flows. An experimenter can select arbitrary combinations of flows offered from multiple campus sites to be injected into her experiment. As we will discuss at length below, maintaining the structural integrity of a traffic stream – such as packet sizes and interpacket timing – may be of primary interest to experimenters, but the packet payload data itself is often not. We will ensure that payload data is opaque to remote experimenters, as requested by those parties responsible for the data. For the moment, the reader should assume the payload is encrypted, or has been masked to 'all zeroes' prior to dissemination.

Available data flows ordinarily fall in two types:

- *Reflected traffic* (non-reactive)
  This traffic is a real-time *mirror* of live data traffic already traversing a campus link. The beginning and end of the mirrored traffic stream is determined by an experimenter's request to initiate and terminate transmission of the desired stream(s), not by any aspect of the campus traffic itself, such as the start or end of a session, transaction, or request-response. The typical experimental use of this traffic will be to serve as bulk external background traffic, redirected to a remote experiment node after traversing one or more testbed container network links. Requested traffic can be optionally tunneled (e.g., GRE).

  Note that this type of traffic is *non-reactive* to congestion caused by other traffic on shared links within the testbed experiment container. Mirrored traffic might also be of interest as foreground traffic within an experiment. For example, an experimenter could be interested in examining the networking characteristics of a flow emanating from a particular scientific instrument, say in the hope of better understanding how to distribute live instrumentation data in the wide-area.

- *Invoked traffic* (reactive)
  The movement of publicly available datasets represent a significant amount of production network traffic flowing between campuses, high performance computing centers (e.g., TACC), and commercial cloud storage services (e.g., Amazon S3). Where available and offered for distribution, STAAS will serve this content for *non-production* distribution purposes to the testbed on behalf of remote experimenters. By a non-production purpose we mean that access to the offered science dataset itself is not the primary science purpose of the distribution, but rather the data traffic generated by requesting it. Invoked on demand and transferred by conventional mechanisms (e.g., GridFTP, http) via SDMZ Data Transfer Nodes (DTNs), the underlying content can be left readable (and, in fact, usable) by the recipient, if desired. Invoked traffic will typically be transferred via TCP, and will consequently be *reactive* to network congestion on shared links within the testbed.

A. Design Goals

The STAAS system seeks to provide a reliable, safe, shared community resource for testbed experimenters by exhibiting the following desirable technical and non-technical properties:

- *Data exchange upon two-sided agreement*: All data traffic that is offered by STAAS is done so with the permission of the parties responsible for the data and its transmission.

These parties might include a campus network operator, the content owner, researchers, or decision makers responsible for data protection. We will discuss more on data privacy, and incentives for data sharing in Sect. V. Traffic only flows into the testbed if explicitly offered by the provider, and explicitly requested by an authorized experimenter. Approved traffic can only be directed to an experimenter's container.

- *Traffic integrity*: To the extent possible the traffic offered to an experiment should maintain the essential characteristics of the traffic at its origin. These characteristics refer to the *structure* of the packet stream (e.g., packet size, rate, timing, duration, etc.) corresponding to a flow.
- *Co-existence with campus data traffic*: The system is expected to be a good citizen on each campus or end system. STAAS traffic using campus network resources should not displace or noticeably degrade other production campus or campus border traffic. That is, simply because a flow is offered by a site does not ensure its continuous availability to an experimenter, if local campus traffic would be adversely affected by serving such a flow. The specific means of prioritizing campus traffic will depend on specifics associated with the network location of offered traffic sources, and must be left up to campus network administrators. Control mechanisms to limit aggregate STAAS packet flow rates might include enforcing traffic caps, local policy routing, router QoS queue controls, DiffServ code points, etc.
- *Traffic realism*: STAAS strives to provide individual and aggregate streams that are reasonably realistic. Of course, if an experimenter demanded complete realism they could elect to test their experiment on an open rather than closed testbed (e.g., GENI). A plausible goal for realism for aggregate traffic to more closely resemble the traffic on a representative public internet link than the experimenter could readily provide through conventional traffic emulation methods such as replay and synthesis.
- *Data privacy*: The release of STAAS traffic header and payload data is intended to be appropriate, harmless and consistent with all applicable laws and regulations governing privacy and data protection. Our intent is to implement appropriate practices for the coordinated protection of science dataflows as suggested in current secure SDMZ data handling recommendations including [10], [11]. We do not underestimate the complexity of handling data traffic appropriately; we take as a significant research thrust to improving the research community's understanding of best science data protection and privacy practices. However, we stress that we are most interested in harvesting the 'low hanging fruit' of widespread, relatively insensitive traffic types. For example, much campus traffic is already encrypted, and other unencrypted "science" traffic (e.g., public datasets being disseminated) might require little or no obfuscation of any kind. Of course, in nearly all cases STAAS will de-identify and modify header data, and trim, mask and modify payload data at lines rates as needed. No administrative campus data, campus network operational data or metadata (e.g., DNS data), personally identifiable data, data involving human subjects without their consent (e.g., brain imaging data), or otherwise sensitive information will be released. An important outcome of this project is developing documents and software artifacts describing and implementing best practices for the appropriate manipulation, obfuscation, protection and transfer of campus data.

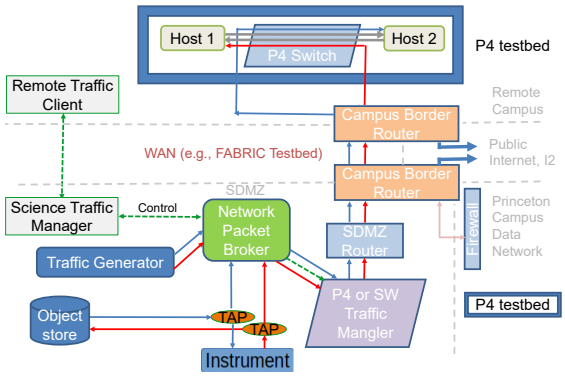

Fig. 2: STAAS architecture on a wide area testbed. In one illustrative use case, tapped bidirectional data flows from a scientific instrument to a local data store on the Princeton campus (red and blue) are directed to a remote site's P4 testbed. The experimenter can study the interaction of these mirrored flows with locally generated traffic between Hosts 1 and 2 (gray).

III. Implementation

Fig. 2 depicts the major hardware and software elements of a STAAS system. Each participating site independently determines what traffic it would like to capture and offer to remote experimenters on-demand. Capturing and filtering campus flows requires deploying the following computing and networking equipment.

- *Passive optical Test Access Points and Port Mirrors* Passive optical Test Access Points (TAPs) are placed in-line on network links bearing traffic to be offered. Each TAP passes all the network traffic it sees and also forwards a copy to a traffic controller such as a *Network Packet Broker*. TAPs accurately retain the integrity of flow characteristics including packet size and timing, and ensure that all traffic arrives to the upstream broker. The unobtrusiveness of fiber TAPS permits safe insertion in campus data links (e.g., between a scientific instrument and a nearby object storage or compute server) even in production settings where instrument operators might ordinarily be extremely cautious about service disruption. The STAAS prototype uses Ixia TAPs [12], and also supports – less preferentially – traffic duplication via switch-based Port Mirrors (PMs) or Switched Port Analyzer (SPAN) ports.

- *Network Packet Broker*
  An NPB provides a simple way to receive, filter/manipulate, aggregate and forward tapped traffic based on policies. Traffic handling policies are jointly set by each campus site and remote experimenter requests. An NPB can be invaluable for protecting data privacy; it can dynamically obfuscate data payloads through *trimming* and *masking* actions that can be performed at line rate while maintaining traffic integrity. Our prototype (Fig. 3) uses the Arista *DANZ Monitoring Fabric* (DMF) – formerly Big Switch Network's *Monitoring Fabric* (BMF) [13], which is also the campus' network-wide monitoring fabric technology. Each STAAS site may use its own campus monitoring fabric for ease of access to redistributable flows. Our prototype fabric is a collection of Dell S41xx ONIE ethernet switches and associated OpenDaylight SDN controller. High speed payload manipulations are performed by a dedicated appliance, an Intel r610 1U server with quad 10Gbs NIC implementing functions in software using the Data Processing Development Kit (DPDK).
- *Programmable Packet Manglers*
  The fabric forwards tapped network traffic through a packet mangler implemented on a (local) programmable hardware or software network switch. The mangler is responsible for packet header processing prior to campus egress, including all aspects of de-identification and forwarding to the requested destination. Programs written in the *P4* language [14] (hardware switch) or linux *nftables* (software switch) manipulate packet header fields, including MAC addresses, IP addresses, time-to-live, checksums, and TCP/UDP source and destination ports. Basic P4-based *packet mangling* functions have already been implemented by the Princeton team in preliminary work such as the *Online Traffic Anonymization System* (ONTAS) [15].
- *Science Traffic Manager*
  A site's Science Traffic Manager is a web-based portal enabling remote experimenters to request, modify, and receive offered traffic. A Traffic Manager comprises two services. A traffic *Advertisement Service* offers available flows, authenticates users, and accepts experimenter requests. A traffic *Controller Service* manages redirecting a requested flow to the specified remote endpoint, and establishes and configures the packet processing pipeline (i.e., flow capture, forwarding, header and payload processing, etc.) to service the request. The Traffic Manager portal issues commands to control traffic workflow by 1) sending destination targets for requested traffic to the Packet Mangler; and 2) issuing RESTful API commands to the NPB via a Python Web Server Gateway Interface (WSGI).
- *Remote Flow Terminators*
  Flow Terminators are virtual or physical equipment that represent endpoints for requested traffic flows at destination networks. The endpoint is specified by the requesting experimenter. As an example, a programmable switch on a destination campus might be a traffic terminus; a P4 program executing on the remote switch can intercept the incoming flow and simply drop all packets using match-action operations. Many other Termination types are possible as well, including simple discard servers (i.e., UDP and TCP port 9) on computing endpoints [16].
- *High performance traffic generators*
  To support experiment repeatability, open source traffic generation software will enable experimenters to select synthetic traffic as a supplement to – or in place of – redirected live traffic. While in principle an individual experimenter could instrument her own experiment with generators, maintaining the integrity of flows at rates up to and exceeding 100 Gbps requires considerable expertise and care [17]. Our prototype is currently using generators such as the *Cisco Realistic Traffic Generator* (TREX) [18].

Fig. 3: A prototype rack implementation (rear view). From top: a 48 and 12 port 10/100G ethernet switch NPB access fabric; 3 1U servers functioning as a) an appliance performing payload obfuscation, b) a software packet mangler and traffic manager, and c) an NPB controller and traffic generator; and a control network switch.

IV. Operation on a Wide-Area Testbed

Fig. 2 depicts how the STAAS system can be deployed and used on a testbed such as FABRIC. Blue lines represent data paths between system components. Dashed green lines depict control traffic paths used to initiate and terminate a desired STAAS data flow. Suppose a remote campus experimenter conceives an experiment that transmits traffic between the P4 testbeds (shown in red), and seeks to also add background traffic traversing WAN testbed links between campuses.

Princeton has advertised available on-demand traffic streams that are externally visible on its *Science Traffic Manager* server portal from a *Traffic Client* browser at the remote site. Advertised flows available for distribution include two types of tapped scientific instruments, synthetic traffic from a traffic generator, and traffic associated with a public dataset on a

local mirror (dark blue rectangles). The experimenter selects one or more traffic streams and specifies a destination target address; other optional, specified properties might include desired packet time-to-live (TTL) value, flow duration, payload obfuscation, etc. Suppose the experimenter elects to initiate the transfer of a tap of a science instrument. The Science Traffic Manager responds to an authorized request by constructing a packet pipeline to fulfill the request. The Manager issues a request (e.g., gRPC or RESTful web API) to the NPB to perform payload processing according to a specified data privacy policy, and to forward the requested tapped traffic to the Packet Mangler. The Manager would request the Mangler perform the necessary header manipulation (e.g., destination IP address overwriting, etc) on the arriving stream for forwarding to the remote site.

Packets departing the Mangler are forwarded through an SDMZ router to a campus border router and on to the target testbed container. At the destination campus the flow exits the testbed and enters the campus network, where packets are forwarded to the experimenter's indicated destination(s) and ultimately discarded.

Next let's consider an example of a specific experiment in a FABRIC testbed setting. Fig. 2 shows a use case designed to carry a scientific instrument's traffic to a remote site's P4 testbed. The remote testbed has two computers Host 1 and Host 2 connected by a P4 switch. Bidirectional traffic (gray) flows between theses hosts. Suppose that the experimenter seeks to understand the effect of the instrument's traffic on this inter-host traffic. The experimenter requests the tapped instrument's bidirectional traffic as two separate unidirectional flows. The traffic filtered as necessary by the Packet Broker (e.g., payload obfuscation via a masking operation), and then processed by the Packet Mangler for forwarding. At the remote testbed the mirrored unidirectional flows each arrive at a distinct P4 switch port, and traverse the switch along the same paths as the inter-host traffic. In the figure the instrument egress traffic in red (ingress traffic in blue) shares the same switch path as foreground traffic from Host 2 to Host 1 (Host 1 to Host 2).

Note of course that any requested flow must traverse multiple campus and backbone switches, routers and shared communication links before arriving to its destination. Each hop potentially degrades the timing integrity of the original captured flow. Our goal is to characterize and quantify these transmission affects, and design overall STAAS system operation to mitigate their impact. Indeed, this understanding if of particular importance for another use case – where the focus of an experimenter's attention is the behavior of the traffic emitted from the instrument itself, not merely its use as background traffic.

There are two properties that make FABRIC a particularly attractive WAN testbed to support STAAS. The first is that the FABRIC edge node contains many of the required elements to support the STAAS system, including a P4 switch, an array of compute servers, high performance programmable NICs, as well as additional components. Use of this equipment can

```
18:05:15.71 IP (proto UDP (17), length 50)
  10.43.233.172.40336 > 10.43.233.171.1788:
  [udp sum ok
    0x0000:  E..2/.@.@.#s.+..
    0x0010:  .+.......a'the.
    0x0020:  password.is.MAGI
    0x0030:  C.

18:13:46.26 IP (proto UDP (17), length 50)
  10.43.233.172.56284 > 10.43.233.171.1788:
  [bad udp cksum]
    0x0000:  E..2p.@.@....+..
    0x0010:  .+.......".XXXX
    0x0020:  XXXXXXXXXXXXXXXX
    0x0030:  XX

18:30:15.19 IP (proto UDP (17), length 50)
  10.43.233.172.46207 > 10.43.233.171.1788:
  [truncated-ip - 4 bytes missing!]
    0x0000:  E..2..@.@.7Y.+..
    0x0010:  .+.......J8the.
    0x0020:  password.is.".
```

Fig. 4: A tcpdump listing of UDP packets with STAAS payload obfuscation. The first received packet's payload is unaltered, with text string "the password is MAGIC". The second packet's payload is masked, with each character replaced with the ASCII character 'X'. The third slices (truncates) the string after 15 characters. Tcpdump correctly reports that the second's length is correct while the checksum is incorrect, and that the third packet was truncated.

lower the overall cost of a STAAS campus system. The second attractive element is that the capabilities of a traffic generation system scales with the number of edge nodes, and FABRIC is anticipated to have dozens of such edge nodes. Note that an alternative approach to traffic ingress to the FABRIC system is also possible – using its connections to public compute clouds as a potential source of live traffic on-demand. However, the costs and administrative overhead associated with this hybrid cloud approach [19] make it prohibitively expensive, due in large part to steep cloud data egress fees.

V. Use Policies, Data Privacy & Security, and Participation Incentives

Achieving testbed traffic realism at scale without carrying production traffic has led us to propose the re-direction of 'live' data streams. We believe this repurposing of science data *traffic* – rather than science *data* – has yet to be fully considered by the research community. The exploration of the appropriate use and practice of repurposing networked science flows is a research topic on its own merits. We are exploring fundamental questions about the ownership, use, regulations, and best practices of science data traffic re-use.

Generally speaking, using science data explicitly shared by others for *secondary use* is considered 'low risk' depending on the nature of the data [20]. Risk factors include the sensitivity of the data, the nature of agreements with human

participants (if any) in the original investigation, and the potential presence of personal identifiers. Personal identifiers suggests particular care in the secondary use of data types such as medical images or media streams involving human subjects. Fortunately, secondary use does not require public (or even encrypted) data – restricted access data can be appropriate for such use. Among our goals is to begin to clarify these topics for both Institutional Review Boards (IRB) and the broader science research community. While a broad range of data types are interesting to study, our project initially focuses on those campus traffic flows that represent secondary use of 1) non-sensitive, de-identified data previously exempted from IRB review, and 2) data specifically authorized by data owners.

What incentives do scientists and others responsible for data transfer have to to share data with research infrastructure experimenters? Preliminary anecdotal feedback from campus domain scientists and instrument operators suggests a strong willingness to support their research colleagues to develop next generation infrastructures by participating in a low-overhead, tightly controlled data sharing collaboration. Scientists responsible for data specify the terms of data sharing, as consistent with their existing data use and privacy agreements. For example, a data owner may request that data from a live instrument is partly or entirely overwritten before external distribution (see Fig. 4). That said, data owners often lack experience with sharing captured live data, as opposed to later publishing a complete, well understood dataset upon project completion. Developing best practices to educate data owners who express sharing concerns will remain a key, ongoing element of our work.

VI. Conclusion

Experimentation on wide area networking testbeds can be very challenging. Experiment operation and behavior across geographies is difficult to observe, measure, diagnose and debug. More sophisticated tools must be developed to provide new forms of experiment support, instrumentation, measurement, and monitoring. As we deploy more 100-1000 Gbps testbed links, experimenters can no longer reasonably be asked to independently, safely, and efficiently create test traffic that provides the realism that their research will demand.

STAAS seeks to organize a collective of testbed-connected institutions to develop and deploy a comprehensive, decentralized, infrastructure-wide traffic distribution service whose performance, flexibility, service offerings and capacity exceed that which could be achieved by individual or small groups of testbed experimenters working in isolation [21]. We believe that this approach will elevate data traffic generation and distribution to a first class experimental testbed resource, allowing researchers easier and faster experiment preparation and execution.

References

[1] "FABRIC: Adaptive Programmable Research Infrastructure for Computer Science and Science Applications)", *https://whatisfabric.net*, 2022.

[2] "PlanetLab." *https://planetlab.cs.princeton.edu*.

[3] M. Berman, J. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, I. Seskar, "GENI: A Federated Testbed For Innovative Network Experiments," *ComNets*, vol. 61(0), 2014.

[4] "AL2S: Internet2's Advanced Layer 2 Service", *https://www.internet2.edu/products-services/advanced-networking/layer-2-services/*.

[5] F. Hermenier, R. Ricci, "How To Build a Better Testbed: Lessons From a Decade of Network Experiments on Emulab," *Tridentcom*, June 2012.

[6] "Network Traffic Generation: A Survey and Methodology," O. Ade Adeleke, N. Bastin, D. Gurkan, *ACM Computing Surveys*, Vol. 55 Issue 2, 2023.

[7] K. V. Vishwanath, A. Vahdat, "Realistic and Responsive Network Traffic Generation," *ACM SIGCOMM 2006*.

[8] P. Barford, M. Crovella, "Generating representative web workloads for network and server performance evaluation", *MMCS*, (1998), pp. 151–160.

[9] R. Harrison, Q. Cai, A. Gupta, J. Rexford, "Network-Wide Heavy Hitter Detection with Commodity Switches," *SOSR'18*, 2018.

[10] V. Nagendra, V. Yegneswaran, P. Porras, "Securing Ultra-High-Bandwidth Science DMZ Networks with Coordinated Situational Awareness", *ACM HotNets XVI*, 2017.

[11] V. Nagendra, V. Yegneswaran, P. Porras, S. Das, "Coordinated dataflow protection for ultra-high bandwidth science networks", *Proceedings of the 35th Annual Computer Security Applications Conference (ACSAC '19)*, NY, 2019.

[12] "Network TAPs, Regenerator TAPs, and TAP Aggregators," *https://www.ixiacom.com/products/network-taps-regenerators-and-aggregators*.

[13] "Big Switch Monitoring Fabric," *https://https://www.arista.com/en/bigswitch/products*.

[14] "P4 Language Consortium," *https://www.p4.org*.

[15] H. Kim, A. Gupta, "ONTAS: Flexible and Scalable Online Network Traffic Anonymization System," *NetAI'19: Proceedings of the 2019 Workshop on Network Meets AI & ML*, August 2019, pp. 15–21.

[16] J. Postel, "IETF RFC 863: Discard Server", *https://tools.ietf.org/html/rfc863*, 1983.

[17] US Dept. of Energy ESnet, "Science DMZ: Data Transfer Nodes : 100G DTN," *http://fasterdata.es.net/science-dmz/DTN/100g-dtn/*.

[18] "Cisco Realistic Traffic Generator," *https://trex-tgn.cisco.com/*.

[19] J. Brassil, I. Kopaliani, "CloudJoin: Experimenting at scale with Hybrid Cloud Computing," *2020 IEEE 3rd 5G World Forum (5GWF)*, 2020, pp. 467-472.

[20] Databrary, "Working with your IRB: Types of Data sharing" *https://www.databrary.org/resources/policies/work-with-irb.html*, retrieved Dec. 2019.

[21] L. Landweber, "2022 MERIF Workshop on Future Midscale Experimental Research Infrastructure (MERI)", *https://sites.google.com/a/us-ignite.org/merif-workshop-2020/2022-agenda*.