Arpit Gupta

Research Interests

As a systems researcher, I design and build easy-to-use and scalable systems that solve real-world problems at the intersection of networking, cybersecurity, and data science.

Education

Summer 2018 (Expected) **Princeton University**, *Ph.D.*, Computer Science.
Advisor: Nick Feamster
How can networks run themselves? My dissertation research centers around answering this question. In particular, to satisfy the increased security, availability, and performance requirements of modern networks; my goal is to design and build self-driving networks that automatically (without humans in the loop) make holistic (not protocol-specific) control decisions in real time.

Spring 2013 **NC State University**, *M.S.*, Computer Science.

Spring 2009 **Indian Institute of Technology, Roorkee**, *B.Tech.*, Electronics & Comm..

Publications

Conferences

[1] **Arpit Gupta**, Rob Harrison, Ankita Pawar, Marco Canini, Nick Feamster, Jennifer Rexford, and Walter Willinger. Sonata: Query-Driven Network Telemetry. *Under Submission.*

[2] Robert MacDavid, Rüdiger Birkner, Ori Rottenstreich, **Arpit Gupta**, Nick Feamster, and Jennifer Rexford. Concise Encoding of Flow Attributes in SDN Switches. In *ACM Symposium on SDN Research (SOSR)*, 2017.
Best Paper Award (1 out of 77).

[3] **Arpit Gupta**, Robert MacDavid, Rüdiger Birkner, Marco Canini, Nick Feamster, Jennifer Rexford, and Laurent Vanbever. An Industrial-Scale Software Defined Internet Exchange Point. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2016.
Community Award (1 out of 255).

[4] Hyojoon Kim, Joshua Reich, **Arpit Gupta**, Muhammad Shahbaz, Nick Feamster, and Russ Clark. Kinetic: Verifiable Dynamic Network Control. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2015.
60 citations till Nov 2017 based on Google Scholar.

[5] **Arpit Gupta**, Laurent Vanbever, Muhammad Shahbaz, Sean Patrick Donovan, Brandon Schlinker, Nick Feamster, Jennifer Rexford, Scott Shenker, Russ Clark, and Ethan Katz-Bassett. SDX: A Software Defined Internet Exchange. In *ACM SIGCOMM*, 2014.
210 citations till Nov 2017 based on Google Scholar.

[6] **Arpit Gupta**, Jeongki Min, and Injong Rhee. Wifox: Scaling wifi performance for large audience environments. In *ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, 2012.
50 citations till Nov 2017 based on Google Scholar.

320 Sherrerd Hall – Princeton, NJ
arpitg@cs.princeton.edu • *www.cs.princeton.edu/~arpitg* • *agupta13*

Workshops & Short Papers

[7] Rob Harrison, Qizhe Cai, **Arpit Gupta**, and Jennifer Rexford. Network-wide heavy hitter detection with commodity switches. *Under Submission.*

[8] Xiaohe Hu, **Arpit Gupta**, Aurojit Panda, Nick Feamster, and Scott Shenker. Preserving Privacy at IXPs. *Under Submission.*

[9] Rüdiger Birkner, **Arpit Gupta**, Nick Feamster, and Laurent Vanbever. SDX-Based Flexibility or Internet Correctness?: Pick Two! In *ACM Symposium on SDN Research (SOSR)*, 2017.

[10] **Arpit Gupta**, Rüdiger Birkner, Marco Canini, Nick Feamster, Chris Mac-Stoker, and Walter Willinger. Network Monitoring as a Streaming Analytics Problem. In *ACM Workshop on Hot Topics in Networks (HotNets)*, 2016.

[11] **Arpit Gupta**, Nick Feamster, and Laurent Vanbever. Authorizing Network Control at Software Defined Internet Exchange Points. In *ACM Symposium on SDN Research (SOSR)*, 2016.

[12] **Arpit Gupta**, Matt Calder, Nick Feamster, Marshini Chetty, Enrico Calandro, and Ethan Katz-Bassett. Peering at the Internet‘s Frontier: A First Look at ISP Interconnectivity in Africa. In *Passive and Active Network Measurement (PAM)*, 2014.
55 citations till Nov 2017 based on Google Scholar.

Professional Experience

2015–Present **Princeton University**, *Research Assistant*, Princeton, NJ.
Mentors: Nick Feamster and Jennifer Rexford
Designed and implemented: a network streaming telemetry system, *Sonata* [10, 1, 7]; and an industrial-scale software-defined Internet exchange platform, *iSDX* [3, 2, 9, 8].

Summer 2016 **Microsoft Research**, *Research Intern*, Redmond, WA.
Mentors: Ratul Mahajan and Monia Ghobadi
Explored the design of a wide-area network controller, *Roshan*, that configures both the optical (physical) and the network layer to make optimal use of limited available resources under failures.

2013–2014 **Georgia Tech**, *Research Assistant*, Princeton, NJ.
Mentor: Nick Feamster
Designed and built a software-defined Internet exchange platform, *SDX* [5], helped with the design and implementation of an event based network management tool, *Kinetic* [4]. Also, analysed multiple active and passive measurement datasets to model ISP interconnectivity in developing regions [12].

2011–2012 **NC State University**, *Research Assistant*, Raleigh, NC.
Mentor: Injong Rhee
Developed *WiFox* [6], that solves the problem of WiFi performance degradation for large audience environments. This technology has been licensed out to Intel.

Summer 2011 **Google**, *Software Engineering Intern*, Mountain View, CA.
Mentor: Nandita Dukkipati
Worked on quantifying the effect of TCP timeouts on Google's search traffic. Instrumented the TCP stack for Google's front end servers to collect the data required for this measurement study.

Spring 2010 **Indian Institute of Science**, *Project Assistant*, Bangalore, India.
Mentor: Anurag Kumar
Designed and implemented a WiFi AP based scheduling algorithm ensuring fairness to clients with disparate link qualities.

Awards

2017 Best Paper Award winner, ACM SOSR
2017 Facebook Fellowship finalist
2016 Community Award winner, USENIX NSDI
2016 Juniper/Comcast SDN Throwdown winner
2015 Facebook Fellowship finalist
2013 Internet-2 Innovation Award winner
2013 Meissner Fellowship, Purdue University
2010 College of Engineering Fellowship, North Carolina State University

Presentations

Sonata: Query-Driven Streaming Network Telemetry

Conferences: ACM HotNets (11/16), NANOG 70 (05/17), P4 Workshop (05/17)

Industry: Comcast (12/16), NIKSUN Inc. (06/17), AT&T (10/17)

Universities: New England Networking & Systems Day, Boston University (10/16)

iSDX: An Industrial-Scale Software Defined IXP

Conferences: USENIX NSDI (03/16), USENIX ATC (06/16), GENI Network Innovators Community Event (12/16)

Industry: AT&T (10/15), Project Endeavour (10/15), Corsa (11/15), CloudRouter (01/16), Open Networking Foundation Webinar (04/16), Appfest (05/16)

Universities: Networked Systems Laboratory, USC (08/15))

Authorizing Network Control at Software Defined IXPs

Conferences: ACM SOSR (03/16)

Industry: Verisign Inc. (08/15)

SDX: A Software Defined Internet Exchange

Conferences: ACM SIGCOMM (08/14), GENI Engineering Conference 20 (06/14), NANOG 59 (10/13), OpenIX Summit (04/15))

Industry: Facebook Inc. (08/14), Microsoft (08/14)

Universities: NetSeminar, Stanford University (10/14)

Peering at the Internet‘s Frontier

Conferences: Workshop on Passive and Active Measurements (03/14)

WiFox: Scaling WiFi Performance for Large Audience

Conferences: ACM SIGCOMM CoNEXT (12/12)

Universities: Duke University (10/12), UNC Chapel Hill (10/12)

Professional Activities

External Reviewer

NSDI 2014, ICNP 2016, SIGCOMM 2017, IEEE/ACM Transactions on Networking, IEEE Transactions on Mobile Computing, Computer Networks, Network Management

Program Committee

Workshop on Self-Driving Networks, SIGCOMM 2018

320 Sherrerd Hall – Princeton, NJ
arpitg@cs.princeton.edu • *www.cs.princeton.edu/~arpitg* • *agupta13*

Panelist

GENI Network Innovators Community Event 2016, CITP Conference on Global Internet Interconnection 2016

Teaching and Mentoring Experience

Teaching Assistant

Spring 2016 Computer Networks (COS 461), Princeton University

Summer 2013 Computer Organization & Assembly Language, (CSC 236), NC State University

Fall 2012 Internet Protocols (CSC 573), NC State University

Course Development

Spring 2016 Securing Cyberspace with Big Data (COS 598E), Princeton University

Summer 2015 Software Defined Networking, Coursera

Fall 2014 Software Defined Networking (CS 4270), Gerogia Tech

Summer 2014 Software Defined Networking, Coursera

Guest Lecture

Fall 2017 Computer Networks (COS 561), Princeton University

Spring 2017 Computer Networks (COS 561), Princeton University

Mentoring

Fall 2017 – * David Liu, Ph.D.

Summer 2017 – * Bridger Hahn, M.S.

Fall 2016 – * Rob Harrison, Ph.D.

2015 – 2017 Rüdiger Birkner, M.S.

2015 – 2016 Robert MacDavid, M.S.

References

Prof. Jennifer Rexford
Department of Computer Science
Princeton University
35 Olden Street, CS 306
Princeton, NJ 08540
jrex@cs.princeton.edu

Prof. Nick Feamster
Department of Computer Science
Princeton University
310 Sherrerd Hall
Princeton, NJ 08540
feamster@cs.princeton.edu

Prof. Ethan Katz-Bassett
Department of Electrical Engineering
Columbia University
500 West 120th Street
New York, NY 10027
ethan@ee.columbia.edu

Dr. Walter Willinger
NIKSUN Inc.
457 N Harrison St
Princeton, NJ 08540
wwillinger@niksun.com

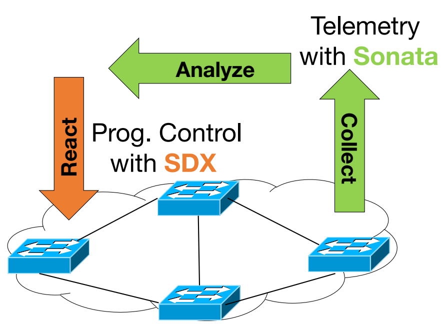
Research Statement

Arpit Gupta

As a systems researcher, I design and build easy-to-use, scalable, and deployable systems that solve real-world problems at the intersection of **networking**, **cybersecurity**, and **data science**. First, I identify real-world problems that network operators and vendors face while working "in the trenches" and understand their practical constraints. I then design innovative and provable solutions to these problems using various theoretical techniques (*e.g.*, optimization, graph theory, machine learning) and emerging technologies (*e.g.*, programmable data plane, scalable stream processors). Finally, I build practical systems with minimal deployment overhead. My systems and the software packages I wrote are widely used in both academia and industry. For example, my software-defined Internet exchange point (IXP) prototype, which won the **Community Award** at *USENIX NSDI* in 2016, is used by many IXP operators across the globe. Also, the recently released, "network streaming telemetry" prototype, is currently being used by network operators and researchers at AT&T.

How can networks run themselves? My dissertation research centers around answering this question. In recent years, with the proliferation of networked devices (e.g., Internet-of-Things), systems (e.g., container networks), and applications (e.g., augmented or virtual reality), the complexity of managing networks at scale has increased significantly. Conventional network management tools and practices built for much simpler networks are ill-suited to handle this increased complexity. To satisfy the increased security, availability, and performance requirements of modern networks; my goal is to design and build **self-driving networks** that automatically (without humans in the loop) make holistic (not protocol-specific) control decisions in real time.

To mitigate the effects of dynamic uncertainty, self-driving networks need to "sense" (*i.e.*, collect **high-velocity** network data), "infer" (*i.e.*, analyze collected **high-volume** data), and "actuate" (*i.e.*, react to different inferred network events) by themselves; that is, they rely critically on closed-loop feedback. My dissertation research focuses on the expressiveness, scalability, and correctness of two key building blocks for this closed feedback loop:



Feedback loop for self-driving networks.

- *Telemetry* [2,4,7], which refers to the process of collecting and analyzing the raw network data to discover various network activities of interest (*e.g.*, inferring presence of DDoS attacks and link failures) in real time; and
- *Programmatic Control* [1,3,5,6,9,10], which refers to the process of programmatically applying the fine-grained reactive actions without disrupting networks' default behavior (*e.g.*, redirecting traffic from congested peering links without creating any forwarding loops).

Telemetry with *Sonata*

Problem. For networks to run themselves, they need to monitor a wide range of network activities. For example, they need to concurrently detect whether the network is under attack and also determine whether there is a device failure in the network. This involves extracting multiple features from the traffic data and combining them to infer activities of interest in real time. Most existing real-time telemetry systems are either not expressive, *i.e.*, they support an insufficient set of telemetry tasks, or they are not scalable, *i.e.*, they fail to scale as the traffic volume intensifies and the number of telemetry tasks increases. In contrast, self-driving networks require running *multiple expressive* telemetry queries over *high-volume* networks. My research focuses on building a distributed real-time network telemetry system that is both *expressive* and *scalable*.

Observation. Many existing telemetry solutions take advantage of either scalable stream processing or programmable data-plane targets for network telemetry—but not both. While at first glance, these two technologies seem inherently different, they both apply a sequence of transformations over packets (tuples). This similarity provides new opportunities to combine their strengths. Also, since many telemetry tasks, the fraction of relevant traffic is typically tiny. Thus, unlike existing task/query-agnostic systems, it makes sense to build query-driven telemetry systems. Based on these observations, I developed an expressive query-driven streaming telemetry system, *Sonata* [2,4], that scales query execution by making use of both streaming and programmable data-plane targets and letting the output of queries themselves drive further processing.

Challenges and Solutions. Ideally, one would like to execute all queries in the data plane itself, but limited data-plane resources (*e.g.*, memory) restricts which portions of the queries to execute in the data plane. Requiring network operators to decide how to partition each query and configure the data-plane targets to run the partitioned queries by themselves can be overwhelming. To address this problem, Sonata provides a declarative query interface where network operators express queries for a range of telemetry tasks as a sequence of dataflow operations over packet tuples. Internally, Sonata takes these queries, representative workload data, and various data-plane constraints and determines the query partitioning plan that minimizes the workload at the stream processor. Sonata's query planner also determines a query-specific refinement plan that iteratively zooms-in over portions of traffic that satisfy the queries, making the best use of limited data-plane resources.

As a first step, I focused on a single-switch implementation for Sonata [2,4]; I'm currently extending this work to build a network-wide telemetry system. For the network-wide settings, Sonata's query planner needs to determine, among other issues, where to evaluate each query; how to partition the stateful operations over multiple hops while making the best use of limited resources along a path; and how to select the thresholds for filter operations [7].

Impact. Building an expressive and scalable telemetry system is an important step towards making self-driving networks a reality. This project has generated significant interest in self-driving networks, with two recently announced workshops (NSF and ACM SIGCOMM) focusing exclusively on this topic. I have also open-sourced this project, and I am currently working with researchers at ISPs (AT&T) and security vendors (NIKSUN Inc.) to express and scale queries for various real-world telemetry tasks with Sonata.

Programmatic Control with *SDX*

Problem. After determining the reactive actions with telemetry, a self-driving network needs to apply these fine-grained actions in the data plane, usually in multi-domain settings. My research focuses on applying programmatic control for the inter-domain networks settings, where by default networks use the Border Gateway Protocol (BGP) to exchange traffic with each other. In my study, I demonstrate that BGP is not suited for applying fine-grained control actions. Ideally, one would like to replace BGP with a clean-slate solution supporting programmatic control at scale. However, the existing Internet-wide deployment of BGP-speaking routers makes such an approach impractical. Thus, the goal of my research was to design and implement a dirty-slate solution that ensures maximal impact with minimal deployment overhead while safely interoperating with BGP.

Observation. In recent years, we have seen the emergence of Internet exchange points (IXPs). They provide a common switching fabric for various networks for exchanging traffic with each other and are strategically located to influence a significant portion of the Internet's inter-domain traffic. I proposed building a software-defined exchange (SDX) [6] that replaces the conventional switching fabric with programmable switches at IXPs. While simple in theory, my research shows that building a practically deployable SDX requires striking a delicate balance between expressiveness, correctness, privacy, and scalability.

Challenges and Solutions. Though SDX enables flexibility at the IXP, everyone else is still using BGP. Thus, it needs to ensure that this additional flexibility does not come at the cost of creating forwarding loops in the network. Requiring SDX participants to express their control program while guaranteeing forwarding correctness can be cumbersome, as they not only have to worry about their own control program but also their peers' programs. To address this problem, SDX provides a virtual switch abstraction to each participant which they can use to express their policies without worrying about others. SDX augments participant's programs to ensure correctness. It then composes them together and compiles the composition to forwarding table entries for the data plane. While such an augmentation ensures correctness, the cost of additional forwarding rules in the data plane can create scalability problems. I developed novel "attribute encoding" schemes [5,6] that make use of several existing protocols and mechanisms to encode the forwarding attributes in the header of each packet before it enters the switching fabric—reducing the number of forwarding table entries required in the data plane.

To ensure that SDX participants can trust the computations at the IXP, I designed and implemented *SGDX* [8], that uses trusted execution environments such as Intel SGX to keep participants' policies private. To ensure that SDX participants can only influence the traffic that they are authorized to over shared physical switch, I built *FLANC* [3], which authorizes action requests over a shared switch for each participant. Finally, to stop participants at multiple IXPs from expressing conflicting policies, I developed *SIDR* [1], which makes it easier for participants to trade expressiveness (*i.e.*, fine-grained control policies) with privacy (*i.e.*, share control policies with other SDXs) to ensure correctness.

Impact. My SDX paper [6] had a significant impact on the research community. It was the first project to demonstrate how programmable switches can enable flexible inter-domain routing at scale. It inspired NSF's first workshop on SDX in 2014, which was followed by multimillion-dollar grants for research in this area. The work also inspired the creation of Endeavour, a multi-university consortium in Europe for SDX-related research. To date, my SDX paper [6] has over 210 citations making it one of the **highest cited** papers from *SIGCOMM 2014*.

I open-sourced the *iSDX* [5] project with the Open Networking Foundation (ONF) and worked closely with two full-time software developers to make the *iSDX* code production ready. I also worked with various network operators, switch vendors, and service providers to help deploy SDX at an inter-government agency exchange and many IXPs across the globe. Project *Endeavour* currently uses *iSDX* as the default platform for their projects. For these efforts, the project won the **Community Award** at *USENIX NSDI 2016*. Also, our *PathSets* [10] paper, that demonstrates how iSDX's encoding scheme are applicable for more general network settings, won the **Best Paper** award at *SOSR 2017*.

Future Directions

Sonata and *SDX* are only two basic building blocks for self-driving networks. In the future, I would like to build upon these efforts to make the distant dream of self-driving networks a reality.

Co-designing Machine Learning and Query Planning Algorithms

Self-driving networks need to train learning models to discover activities of interests. Currently, learning algorithms only focus on training the learning models to maximize accuracy, and query planning algorithms solely concentrate on minimizing the execution cost for queries that represent these learning models. This disconnect between the two often results in very accurate yet prohibitively expensive learning models that are not deployable in the production networks.

I observed that it is possible to augment the optimization problem that most state-of-the-art learning algorithms solve to trade accuracy against query execution cost while accounting various resource constraints (*e.g.*, limited memory in the data plane) to determine the learning model as well as the query plan. Based on this observation, I'd like to design and build a telemetry systems, *SonataML*, that co-designs the algorithms for the machine learning and query planning problems to train realistic and deployable learning models.

While the idea appears to be straightforward in principle, just embedding the cost and constraint metrics to the learning problems is not enough. Since the computational complexity of an optimal solution grows exponentially with the number of features for the learning problem, one needs to design heuristic algorithms that can find solutions to this problem that are just "good enough".

Another challenge is the availability of labeled training data. As most of the activities of interests are needles in a haystack, collecting labeled training data to bootstrap *SonataML* is extremely hard. I envision adapting existing active learning algorithms that can leverage the knowledge of the domain experts (security analysts and network operators) to generate as much of labeled data as possible while minimizing the workload for these human experts. As the nature of security and performance problems evolve with time, rather than removing humans from the loop altogether, I envision the active learning to be an intrinsic part of the self-driving networks, where the workload for the human experts reduces over time as the networks become more capable of running the networks by themselves.

Expanding Networking Telemetry Footprint

Self-driving networks require insights from multiple vantage points in the network. So far, *Sonata* focuses primarily on inferring network activities at the core locations of the Internet (*e.g.*, border routers for large ISPs). Although such deployments provide broader visibility with limited overhead, they miss several insights such as end-to-end performance for specific web applications (*e.g.*, Netflix, CNN, etc.) in a region. These insights, often available only at the edge of the Internet (*e.g.*, home routers, web browsers, etc.), are not only critical for robust self-driving networks but are also crucial to answering various policy questions on socio-political issues such as net neutrality, data privacy, censorship, etc.

I believe using cheaply available tools (e.g., programmable NICs and Raspberry Pis) to design a lightweight Sonata, deployed at home networks, can help expand the network telemetry footprint for self-driving networks. Here, in most cases, the entities generating data (producers) and processing data (consumers) are different. For example, quality-of-experience data for latency-sensitive applications are produced by end-users but can be consumed by an eyeball ISP to improve its network's performance. As a result, there is fundamental friction between the two entities, one trying to get as much data as possible and the other trying to preserve their privacy. Thus, the design of such a decentralized platform needs to execute queries in a privacy-preserving manner.

To ensure privacy-preserving query execution, I plan to design: (1) *Expressive User Interface* for end users to express *what* data to share and with *whom*. I plan to collaborate with programming languages and privacy experts to create a domain-specific language to express data sharing policies; (2) *Data Privacy and Integrity* guarantees for data producers and consumers respectively. To ensure data privacy, I plan to extend my previous work (*FLANC* [3]) to employ formal logic tools for generating policy compliance proofs for input telemetry queries. I also plan to expand, *SGDX* [8], to design protocols that make use of trusted execution environment at local nodes for guaranteeing data integrity; and finally, (3) *Model Privacy* guarantees for data consumers. Learning models are the intellectual property of the consumers, and they have good reasons to keep it private. To ensure model privacy, I plan to use secure multi-party computation to design protocols that distribute computations to local nodes such that privacy of learning models is preserved.

Beyond Self-Driving Networks

The computational complexity of machine learning models has grown significantly in recent years. To catch up with this increased computational requirements, most cloud service providers have responded by horizontally scaling the compute clusters. However, rather than simply throwing more resources at the problem, one needs to step back and think more carefully and explore if there are new technologies on the horizon that can make better use of limited available resources.

Programmable data-plane targets provide the opportunity to scale data analytics computations vertically. I believe *Sonata*'s idea of offloading query execution to data-plane targets applies to more general analytics settings. I plan to design and implement a **general-purpose analytics stack** that is not only aware of the compute capabilities and limitations of CPU-based compute nodes but also the set of data-plane targets connecting these nodes.

To make this system a reality, I need first to demonstrate that using data-plane targets for offloading analytics computations is cheaper than using more CPU nodes. Answer what kind of data center topologies will make the best use of programmable data-plane targets? What kind of targets to use, *i.e.*, whether one should deploy cheaper but low throughput targets (*e.g.*, smart NICs), or high throughput but expensive targets (*e.g.*, switching fabrics), or both? Also, whether one should use off-the-shelf programmable data-plane targets or design custom switch architectures tailored to data analytics tasks?

Second, I plan to design a query planner that for a given topology, compute nodes and programmable data-plane targets; determines packet processing pipelines that execute over both the data-plane and the streaming targets. The goal of this query planner is to minimize the execution time while considering various data-plane constraints (*e.g.*, limited memory) and limitations (*e.g.*, cannot perform multiplication or division operation). The query planner also needs to determine the data structures for packet's payload—reducing the parsing overhead for the intermediate data-plane targets. Finally, I'd also like to expand the idea of co-designing the algorithms for learning and query planning by applying it to more commonly-used machine learning problems.

References

[1] R. Birkner, A. Gupta, N. Feamster, and L. Vanbever. SDX-Based Flexibility or Internet Correctness?: Pick Two! In *ACM Symposium on SDN Research (SOSR)*, 2017. (Cited on pages 1 and 3.)

[2] A. Gupta, R. Birkner, M. Canini, N. Feamster, C. Mac-Stoker, and W. Willinger. Network Monitoring as a Streaming Analytics Problem. In *ACM Workshop on Hot Topics in Networks (HotNets)*, 2016. (Cited on pages 1 and 2.)

[3] A. Gupta, N. Feamster, and L. Vanbever. Authorizing Network Control at Software Defined Internet Exchange Points. In *ACM Symposium on SDN Research (SOSR)*, 2016. (Cited on pages 1, 3 and 4.)

[4] A. Gupta, R. Harrison, A. Pawar, M. Canini, N. Feamster, J. Rexford, and W. Willinger. Sonata: Query-Driven Network Telemetry. *Under Submission*, 2018. (Cited on pages 1 and 2.)

[5] A. Gupta, R. MacDavid, R. Birkner, M. Canini, N. Feamster, J. Rexford, and L. Vanbever. An Industrial-Scale Software Defined Internet Exchange Point. In *USENIX NSDI*, 2016. (Cited on pages 1, 2 and 3.)

[6] A. Gupta, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett. SDX: A Software Defined Internet Exchange. In *ACM SIGCOMM*, 2014. (Cited on pages 1, 2 and 3.)

[7] R. Harrison, C. Qizhe, A. Gupta, and J. Rexford. Network-Wide Heavy Hitter Detection with Commodity Switches. *Under Submission*. (Cited on pages 1 and 2.)

[8] X. Hu, A. Gupta, A. Panda, N. Feamster, and S. Shenker. Preserving Privacy at IXPs. *Under Submission*. (Cited on pages 3 and 4.)

[9] H. Kim, J. Reich, A. Gupta, M. Shahbaz, N. Feamster, and R. Clark. Kinetic: Verifiable Dynamic Network Control. In *USENIX NSDI*, 2015. (Cited on page 1.)

[10] R. MacDavid, R. Birkner, O. Rottenstreich, A. Gupta, N. Feamster, and J. Rexford. Concise Encoding of Flow Attributes in SDN Switches. In *ACM Symposium on SDN Research (SOSR)*, 2017. (Cited on pages 1 and 3.)

Teaching Statement

Arpit Gupta

As a graduate student, I have had the privilege to contribute to *eight* courses and mentor *five* graduate students. These experiences have strengthened my resolve to pursue a career in academia where I would have the privilege to not only teach but also advise undergraduate and graduate students.

As a faculty, I am looking forward to opportunities for teaching introductory courses including *computer networks*, *cybersecurity*, and *big-data systems* at the undergraduate and graduate levels. I am also interested in developing interdisciplinary graduate courses that combine the areas of networking, security, and data science.

Lectures. I have given multiple lectures for the undergraduate (Computer Networks [2]) and graduate (Advanced Computer Networks [3]) classes at Princeton. In my lectures, I take great care to excite students about the material and help them discover new interests. For example, in my lectures, rather than speaking exclusively in the abstract, I motivated content-delivery networks with cat videos and used real-world configurations from Princeton's network to explain firewalls and border routing. Unlike other fields, networking continues to evolve at a relatively rapid pace. Thus, an excellent lecture in networking should go beyond textbooks and discuss recent research advancements, showing students how the field is growing and how they can make an impact of their own.

Course Development. I have also contributed to the development of two new courses: *Software-Defined Networking* (SDN) and *Securing Cyberspace with Big Data* [4] at Georgia Tech with Nick Feamster and at Princeton University with Walter Willinger, respectively. The SDN course was offered both in-class [6] and online [5]. The online version, offered over Coursera in Summer 2014 and 2015, drew tens of thousands of students.

Programming Assignments. Given the panoply of careers that students embark upon after graduation, it is crucial to help them develop transferable problem-solving skills. I believe programming assignments are the right medium to hone such skills. I have contributed to the design of multiple programming assignments for the networking courses at Princeton and Georgia Tech.

I try translating cutting-edge research into programming assignments. This approach not only exposes students to latest developments in the area but also enhances the quality of the research itself. For example, my experience designing programming assignments for the *SDX* [10] and the *Sonata* [11] projects forced me to make the programming interface for these systems more intuitive and easy-to-use.

Mentoring. I was privileged to have the opportunity to mentor five graduate students on specific projects. I supervised Robert MacDavid with Jennifer Rexford on the iSDX [8] and PathSets [9] projects. I mentored Rüdiger Birkner from ETH Zürich with Laurent Vanbever and Nick Feamster on the *iSDX* [8], *SIDR* [1], and *Sonata* [7] projects. I am currently mentoring Rob Harrison with Jennifer Rexford on the Sonata [7] project, Bridger Hahn with Nick Feamster on the IoT security project, and David Liu with Nick Feamster and Mike Freedman on co-designing machine learning and query planning algorithms. I have also mentored many junior students: Hooman Mohajeri, Disney Yan Lam, and Laura Roberts; at Princeton.

As a mentor, I try to help students discover their own interests, help them identify problems where they can capitalize on their strengths, and develop a taste for research. For example, with Robert MacDavid (background in theory) and Rüdiger Birkner (background in systems), I worked on the design of the encoding algorithm and the design and implementation of a production-quality prototype, respectively for the *iSDX* project [8]. Giving them a well-defined problem, tailored to their strengths, boosted their confidence to do research and make an impact. For example, Robert applied the techniques we developed for the *iSDX* project to more general settings beyond Internet Exchange Points and that project ultimately won the **best paper** award at ACM SOSR 2017 [9].

I will encourage my students to interact with developers and operators working "in the trenches" to learn about real-world problems. To facilitate this, I will help them find industrial internships, especially during the early years of their graduate studies; and attend operational meet-ups and conferences, *e.g.*, North American network operator group (NANOG) meetings. Among the gamut of operational problems, I will provide constructive feedback to my students helping them to refine their ideas and identify impactful problems that require a more principled solution.

References

[1] R. Birkner, A. Gupta, N. Feamster, and L. Vanbever. SDX-Based Flexibility or Internet Correctness?: Pick Two! In *ACM Symposium on SDN Research (SOSR)*, 2017.

[2] COS 461: Computer Networks, Princeton University, Spring 2016. `https://www.cs.princeton.edu/courses/archive/spring16/cos461/`.

[3] COS 561: Advanced Computer Networks, Princeton University, Fall 2017. `http://www.cs.princeton.edu/courses/archive/fall17/cos561/`.

[4] COS 598: Securing Cyberspace with Big Data, Princeton University, Spring 2016. `https://registrar.princeton.edu/course-offerings/course_details.xml?courseid=002132&term=1164`.

[5] Coursera SDN Course. `class.coursera.org/sdn-002/`.

[6] CS 4270: Software Defined Networking, Georgia Tech, Fall 2014. `http://noise.gatech.edu/classes/cs8803sdn/fall2014/`.

[7] A. Gupta, R. Harrison, A. Pawar, R. Birkner, M. Canini, N. Feamster, J. Rexford, and W. Willinger. Sonata: Query-Driven Network Telemetry. *arXiv preprint arXiv:1705.01049*, 2017.

[8] A. Gupta, R. MacDavid, R. Birkner, M. Canini, N. Feamster, J. Rexford, and L. Vanbever. An Industrial-Scale Software Defined Internet Exchange Point. In *USENIX NSDI*, 2016.

[9] R. MacDavid, R. Birkner, O. Rottenstreich, A. Gupta, N. Feamster, and J. Rexford. Concise Encoding of Flow Attributes in SDN Switches. In *ACM Symposium on SDN Research (SOSR)*, 2017.

[10] SDX Programming Assignment. `https://github.com/PrincetonUniversity/Coursera-SDN/tree/master/assignments/sdn-ixp`.

[11] Sonata Programming Assignment. `https://github.com/Sonata-Princeton/SONATA-DEV/tree/tutorial/sonata/tutorials/Tutorial-1`.

Diversity Statement

Arpit Gupta

I had the opportunity to work under the tutelage of Nandita Dukkipati at Google Inc., Marshini Chetty at Georgia Tech, and Jennifer Rexford at Princeton University. The experience was enlightening as it enriched my perspective for research and advising. It also helped me appreciate the importance of diversity in science and engineering. Unfortunately, given the low representation of women in these fields, I believe mine was a very privileged experience. I am committed to achieving equity and enhancing diversity in science and engineering areas—ensuring that the representation of not only women but also other minority groups is reflective of current times.

Teaching & Advising. As a graduate student, I have had the opportunity to mentor two junior female graduate students. The experience helped me to recognize the barriers faced by women and other minorities engaging in computer science. As a faculty, I would be sensitive to these problems and create an environment of inclusion in my classes and research group where students from different background feel welcomed and can make progress. I will also actively recruit and train female and minority students in my area and help organize mentoring workshops to encourage them to pursue careers in STEM fields.

Volunteering. I will be working as a volunteer to the newly started program, *AI4All*, at Princeton University. The goal of this program is to increase diversity and inclusion in AI development, policy, and research. This year the plan is to host around 20–30 high school students, who belong to different racial minorities, in summer for 2–3 weeks. I plan to contribute to curriculum development for this program by adding a project that makes use of our smart-home lab, equipped with several state-of-the-art IoT devices, that I helped set up with my advisor Nick Feamster at Princeton University.

Currently, the program has an acceptance rate of around ten percent. The low acceptance rate is attributable to the overhead of hosting students for the duration of this program. I believe to achieve the stated goal, improving the outreach of this program is critical. In the future, I plan to use my expertise in designing massive online courses, to develop an online version of this program to reach out to underrepresented student groups in remote locations.

Workforce Retraining. As the technology keeps evolving continuously, re-educating the workforce ensuring that they can catch up with its fast pace is a topic of national (and global) interest in current times. Networking discipline is an excellent example of how the nature of jobs evolves with time. Conventional networking jobs required a deep understanding of multiple complex networking protocols. In contrast, many networking jobs now require expertise in programming languages and software engineering instead.

In the past, I contributed to an online course on software-defined networking. It provided me with an opportunity to reach out to tens of thousands of students from different age groups and with diverse backgrounds. Many of them were mid-career networking employees honing their skills to embrace the changing ecosystem. Being able to contribute to such a massive workforce retraining was a gratifying experience for me. In the future, I plan to continue developing new courses that can contribute to such large-scale workforce retraining efforts.

Research. I grew up in a developing country where I experienced first-hand the social and economic implications of poor Internet connectivity. In my research, I investigated technology and policy issues that contribute to sparse connectivity in the developing regions. More specifically, I conducted a measurement study to understand why the state of interconnections is weak in the African subcontinent. My research answered why the presence of various Internet exchange points (IXPs) in the region is not contributing to dense interconnection as it did in Europe. I worked with researchers from a policy think-tank in South Africa (Research ICT Africa) for this project. Our efforts helped raise awareness about the state of interconnectivity in the region and encouraged adoption of best practices at local IXPs.

For my interconnection study, we used the BISmark tool, deployed at few home networks in South Africa and Kenya. This tool only supports a limited set of telemetry queries—restricting the nature of questions we could answer. In the future, I plan to design a more lightweight version of *Sonata* for home networks that can be widely deployed in developing countries to answer a richer set of queries and help drive policy decisions in the region.