IMPLICATIONS OF THE SOFTWARE DEFINED NETWORKING REVOLUTION FOR TECHNOLOGY POLICY

NICK FEAMSTER*

Communications networks have become increasingly programmable over the last decade. That programmability has facilitated the automation of (1) gathering data from the network, (2) making inferences based on the measured network data, and (3) automatically reconfiguring aspects of the network based on the measurements and inferences. This type of automation promises to revolutionize the ways networks are operated and is already changing the way that large transit and data center networks are operated. The extent to which networks can now be automated nonetheless introduces new challenges for policy, law, and regulation. In the same way that the explainability of automated algorithms will be important for automated decisions writ large, explainability in network automation will also be critical when considering policy and legal issues in future programmable communications networks. This article provides a brief, general primer on software-defined network (SDN) and highlights technology policy challenges that loom on the horizon for technology policy as our communications networks become more automated.

* Nick Feamster is the Neubauer Professor of Computer Science and Director of the Center for Data and Computing at the University of Chicago. Professor Feamster's research focuses on many aspects of computer networking and networked systems, with a particular focus on network operations, network security, and censorship-resistant communication systems. Ph.D. (Computer Science), MIT, 2005; S.B. (Electrical Engineering) and M.Eng. (Computer Science), MIT, 2001.

INTRODUCTION

The last fifteen years has seen a rise in programmable networking technology that has enabled rapid innovation in (1) the policies that control how network traffic is forwarded; (2) the ability to collect meaningful measurements from the network, from low-level statistics to higher-level information about user experience; and (3) the ability to react to changing network conditions in real-time. This article discusses how certain aspects of programmable networking—sometimes referred to broadly as *Software-Defined Networking (SDN)*—may enable new capabilities that could re-shape ongoing policy and legal discussions.

In short, SDN makes the underlying hardware infrastructure that forwards network traffic more flexible. Conventional networking devices (e.g., commodity routers and switches) have typically made it relatively difficult for network operators either to measure network conditions or control network traffic flows. SDN effectively makes network control much more flexible, because a network's behavior can be controlled from a single high-level software *control program*.[1] Over the past decade, we have taken advantage of the capabilities of SDN—both the programmability of the software control, as well as the underlying network hardware—to make it possible for network operators to gain additional flexibility in how different traffic flows are forwarded, as well as how they are able to infer and react to network incidents, from network intrusion to performance degradation, in real time.

[1] Tony Wang, *Benefits and the Security Risk of Software-Defined Networking*, 4 ISACA J. 25, 25 (2016).

This article begins with a brief discussion of recent developments in SDN. We begin with a discussion of the advances in SDN technology and how these technical advances are making it easier for network operators to automate everything from network monitoring to traffic prioritization. We then explore the implications that these new capabilities for monitoring and automated actions have for tech policy. Finally, we explore the relationship of SDN to network virtualization, exploring both how SDN and network virtualization relate to one another, as well as the implications of network virtualization for technology policy.

I. SDN: WHAT'S NEW IN TECHNOLOGY?

SDN describes a type of network design where a software program runs separately from the underlying hardware routers and switches can control how traffic is forwarded through the network.[2] While in some sense, one might think of this concept as "nothing new"—after all, network operators have been pushing configuration to routers with Perl scripts for decades—SDN brings several new twists to the table: (1) high-level languages to control network forwarding behavior; (2) programmable network hardware; (3) scalable, tunable network measurement capabilities; and (4) the ability to "close the loop" with automated decision-making capabilities.

A. Single Program Control of Multiple Networking Devices

The notion that many devices can be controlled from a single software "controller" facilitates coordinated decisions across the network, as opposed to the configuration of each router and switch essentially being configured (and acting) independently. When we first presented this idea for Internet routing in the mid-2000s,[3] it was highly controversial. Some even retorted that SDN was failed phone-company thinking—this centralized controller nonsense could only come from telecommunications enthusiasts, or so-called "bellheads."[4]

Needless to say, the idea is a bit less controversial now. These ideas have taken hold both within the data center—the wide

[2] *Id.*

[3] Nick Feamster et al., *The Case for Separating Routing from Routers*, PROC. ACM SIGCOMM 2004 WORKSHOPS 5.

[4] The Internet is, after all, "decentralized."

area[5]— and at interconnection points.[6] Technology such as the Software Defined Internet Exchange Point (SDX) makes it possible for networks to exchange traffic only for specific applications—video streaming, for example—or to route traffic for different application along different paths.[7] From an operational perspective, technology such as SDX is a clear win. Prior to such technology, it would have been difficult to implement fine-grained traffic control, including those traffic forwarding decisions that are specific to an application. Such specialized, customized traffic control is becoming increasingly easy.

B. Programmable Hardware in Network Devices

Whereas conventional network devices relied on forwarding performed by fixed-function ASICs, the rise of companies such as Barefoot Networks have made it possible for network architects to customize forwarding behavior in the network.[8] Essentially, the technology is making it possible to customize not only coordinated software control over existing networking hardware, but also the hardware itself. This hardware customizability potentially allows network operators with much wider latitude to determine not only how individual network traffic flows are forwarded, but also what kind of information the hardware collects about each traffic flow.

This capability is already being used for designing network architectures with new measurement and forwarding capabilities, including the ability to measure detailed timing information of individual packets as they traverse each network hop, as well as to scale native multicast to millions of hosts in a data center.[9]

[5] *See* Sushant Jain et al., *B4: Experience with a Globally-Deployed Software Defined WAN*, 43 ACM SIGCOMM COMPUTER COMM. REV., Oct. 2013, at 3; *see also* Linda Hardesty, *Google Brings SDN to the Public Internet*, SDXCENTRAL (Apr. 4, 2017. 3:41 PM), https://www.sdxcentral.com/articles/news/google-brings-sdn-public-internet/2017/04/ [https://perma.cc/9Y84-FHNR]; *What Is Software-Defined WAN (or SD-WAN or SDWAN)?*, SDXCENTRAL https://www.sdxcentral.com/networking/sd-wan/definitions/software-defined-sdn-wan/ [https://perma.cc/TF5Y-PN45] (last updated Apr. 2019).

[6] Arpit Gupta et al., *SDX: A Software Defined Internet Exchange,* 44 ACM SIGCOMM COMPUTER COMM. REV., Oct. 2014, at 551, 551; *see also iSDX*, ONF, https://www.opennetworking.org/projects/isdx/ [https://perma.cc/SY8G-XJM3].

[7] Gupta et al., *supra* note 6, at 552–53.

[8] *See* Anders Keitz, *Barefoot Networks Leads Race for Programmable, Fast Chips*, THESTREET (Oct. 12, 2016, 6:30 PM), https://www.thestreet.com/story/13851109/1/barefoot-networks-leads-race-for-programmable-fast-chips.html [https://perma.cc/9C8X-GWTM].

[9] Nick Feamster, *Software-Defined Networking: What's New, and What's New for Tech Policy?,* CIRCLEID (Feb. 12, 2018, 9:40 AM), http://www.circleid.com/posts/20180212_software_defined_networking_what_is_new_and_new_for_tech_policy/ [https://perma.cc/8WHJ-YSJG].

C. *Programmable Network Monitoring Capabilities*

For decades, network operators have faced the challenge of addressing security and performance issues in their networks with extremely limited network measurement tools. At best, these tools could give a glimpse into how much traffic was flowing over any particular part of the network or an approximation of the paths that network traffic might take across the network.[10]

Programmable networks have dramatically improved the capabilities to measure the network. For example, the Sonata streaming network telemetry platform that was built over the last five years allows a network operator to implement network measurements in terms of simple queries in a familiar high-level programming language.[11] Indeed, the last decade has seen a plethora of tools and programming languages to make network measurement easier to express—and more traceable to implement.

Ten years ago, it was much more difficult to perform fine-grained network measurements that were targeted towards a particular device, application, or user. Today and in the future, the increased flexibility and capabilities of emerging network monitoring tools (and the data structures and algorithms that support them) make it possible to ask more fine-grained questions about network traffic.

Programmable network monitoring capabilities present a double-edged sword for privacy. On the one hand, such a level of control may increase privacy risks, because they could allow an operator to perform fine-grained monitoring that is targeted towards a specific device, application, individual, or location. For example, conventional tools make it more difficult to capture traffic that precisely satisfies a specific query (e.g., "Capture all Web requests from smart TVs" in a particular subscriber's home). Programmable network monitoring will conceivably make it far easier to implement this type of targeted monitoring.

On the other hand, programmable network measurement may present several types of opportunities to improve user privacy. One such opportunity would be the ability to implement "conditional" measurement. For example, whereas a conventional network monitoring device might be "all or nothing" in its ability to perform deep packet inspection, a programmable network device could

[10] *See* PACKET DESIGN, UNDERSTANDING LOGICAL NETWORK OPERATIONS WITH ROUTE ANALYTICS AND NETFLOW 4 (2010). Network monitoring software such as Cisco Netflow (standardized as IPFIX) allows a network operator to determine the traffic flows that have traversed a particular router, as well as the amount of traffic in each flow. *Supra*, at 3. In a similar vein, each Internet router makes it possible to determine the routes to each Internet destination by periodically logging a snapshot of the Internet routing table. *Supra*.

[11] Arpit Gupta et al., *Sonata: Query-Driven Streaming Network Telemetry*, PROC. 2018 CONF. ACM SPECIAL INT. GROUP ON DATA COMM. 357, 357.

implement such traffic capture *only* when certain conditions are satisfied—for example, the programmable network device may only capture a complete traffic trace from a device if the domain names, that the device looks up, indicate a possible infection or device compromise.

Another possibility where programmable measurements could improve user privacy is the opportunity to perform more extensive processing and aggregation on network devices at the edge, as opposed to collecting and aggregating potentially private or sensitive data for centralized processing.

Whether programmable monitoring ultimately improves or decreases the privacy of individuals depends on the nature of how it is implemented.

D. Automated Decision Making in Network Management (“AI Meets Networking”)

For years, network operators have applied machine learning to conventional network security and provisioning problems, including the automated detection of spam, botnets, phishing attacks, bulletproof web hosting, and so forth.[12] Operators can also use machine learning to help answer complex “what if” performance analysis questions, such as what would happen to web page load or search response time if a server was moved from one region to another, or if new network capacity was deployed. Much of this work, however, has involved developing systems that perform detection in an offline fashion (i.e., based on collected traces). Increasingly, with projects like Google Espresso[13] and Facebook Edge Fabric,[14] we are starting to see systems that close the loop between measurement and control.

Likely it will not be long before networks begin making these kinds of decisions based on even more complex inputs and inferences in real time to improve aspects of network operations from performance to security. For example, machine learning

[12] *See* Ram Basnet et al., *Detection of Phishing Attacks: A Machine Learning Approach*, *in* 226 STUDIES IN FUZZINESS AND SOFT COMPUTING: SOFT COMPUTING APPLICATIONS IN INDUSTRY 373, 373–74 (Bhanu Prasad ed., 2008); Elaheh Biglar Beigi et al., *Towards Effective Feature Selection in Machine Learning-Based Botnet Detection Approaches*, 2014 IEEE CONF. ON COMM. & NETWORK SECURITY 247, 248; Omar Saad et al*., A Survey of Machine Learning Techniques for Spam Filtering*, 12 INT’L J. COMPUTER SCI. & NETWORK SECURITY 66, 66 (2012); *see generally* Sumayah Alrwais et al., *Under the Shadow of Sunshine: Understanding and Detecting Bulletproof Hosting on Legitimate Service Provider Networks*, 2017 IEEE SYMP. ON SECURITY & PRIVACY 805.

[13] *See generally* Kok-Kiong Yap et al., *Taking the Edge Off with Espresso: Scale, Reliability and Programmability for Global Internet Peering*, SIGCOMM ’17: PROC. CONF. ACM SPECIAL INT. GROUP ON DATA COMM. 432 (2017) (describing “Espresso Google’s SDN-based Internet peering edge routing infrastructure.”).

[14] *See generally* Brandon Schlinker et al., *Engineering Egress with Edge Fabric: Steering Oceans of Content to the World*, SIGCOMM ’17: PROC. CONF. ACM SPECIAL INT. GROUP ON DATA COMM. 418 (2017) (describing Facebook’s Edge Fabric).

techniques are currently being developed to automatically detect the resolution of a video stream in real time. Looking ahead, network control systems may automate not only quality inference but also methods that can improve performance on the fly—for example, by automatically re-encoding the video at a different bitrate or by sending it over a less congested path in the network. Some video streaming providers already implement a version of this type of adaptation, using factors such as network delay to adjust encoding in real time.[15]

II. SDN: WHAT'S NEW FOR TECH POLICY?

The new capabilities that SDN offers presents a range of potentially challenging questions at the intersection of technology, policy, and law. A few of these challenges include: (1) service level agreements, (2) network neutrality and Internet transparency, (3) jurisdictional borders and nation-state concerns, (4) liability and failures, and (5) privacy. The rest of this section discusses these questions in more detail.

A. Service Level Agreements

A common contractual instrument for Internet Service Providers (ISPs) is the Service Level Agreement (SLA).[16] SLAs typically involve guarantees about network performance such as: packet loss will never exceed a certain amount or latency will always be less than a certain amount.[17] SDN presents both new opportunities and challenges for Service Level Agreements. In terms of *opportunities*, SDN allows operators to define more sophisticated traffic forwarding behavior—sending traffic along different paths according to destination, application, or even the conditions of individual links along and end-to-end path at a particular time. This additional flexibility can potentially enable more efficient use of network resources by enabling sophisticated traffic forwarding policies that adapt to changing network conditions. For example, the network could use higher-level information about protocols, including the resolution of the video or the extent to which the video is experiencing rebuffering, and

[15] Chaitanya Ekanadham, *Using Machine Learning to Improve Streaming Quality at Netflix*, MEDIUM: NETFLIX TECH BLOG (Mar. 22, 2018), https://medium.com/netflix-techblog/using-machine-learning-to-improve-streaming-quality-at-netflix-9651263ef09f [https://perma.cc/3GW9-DP5J].

[16] *See* Panita Pongpaibool & Hyong S. Kim, *Providing End-to-End Service Level Agreements Across Multiple ISP Networks,* 46 COMPUTER NETWORKS 3, 3 (2004).

[17] *See Global IP Network SLA*, NTT COMM., http://www.us.ntt.net/support/sla/network.cfm [https://perma.cc/UT9S-NRY9]; *Global Latency and Packet Delivery SLA*, VERIZON, https://enterprise.verizon.com/terms/global_latency_sla.xml [https://perma.cc/9ASG-AYF2].

automatically re-route around links with high utilization that were resulting in poor video quality.

Yet, the opportunity to create these types of complex SLAs also presents new challenges related to enforcement. How does one go about making sure the SLA continues to be enforced when all routing and forwarding decisions become automated and all interconnects look like Google Espresso—where an algorithm is effectively making decisions about where to forward traffic?[18] What new challenges and opportunities do the new capabilities of programmable measurement bring for SLAs?

Some aspects of SLAs are notoriously difficult to enforce today, because they require detailed accounting. For example, an SLA might specify that a customer could achieve a certain level of availability or average loss rate to some set of destinations. Enforcing that type of service-level guarantee requires performing pervasive monitoring across all traffic flows and computing statistics on those flows, typically *post hoc*. If programmability allows ISPs to offer more complex service offerings the extent of monitoring required to verify an SLA may also become more difficult, depending on exactly the nature of what is promised. On the other hand, advancements in network telemetry will also make SLAs easier for customers to validate. There are huge opportunities in the validation of SLAs, and once these become easier to audit, a whole new set of legal and policy questions will arise.

B. *Network Neutrality and Transparency*

Although the Federal Communication Commission's release of the Restoring Internet Freedom Order (the Order)[19] earlier this year effectively rescinds many of the "bright line rules" that we have come to associate with net neutrality (i.e., no blocking, no throttling, no paid prioritization), the Order nevertheless leaves in place many transparency requirements for ISPs. The Order still requires ISPs to disclose any practices relevant to blocking, throttling, prioritization, congestion management, application-specific behavior, and security.[20] As with SLA definition and enforcement, the transparency aspects of the Order may become more nuanced and relevant as SDN makes it possible for network operators to automate network decision-making, as well as for

[18] Based, perhaps, on a long list of features ranging from application Quality of Experience to estimates of user attention, and incorporated into an inscrutable "deep learning" model.

[19] *Restoring Internet Freedom*, WC Dkt. No. 17-108, Declaratory Ruling, Report and Order, & Order, FCC 17-166 (2018), https://docs.fcc.gov/public/attachments/FCC-17-166A1.pdf [https://perma.cc/85UR-YMND].

[20] *Id.*

consumers to audit some of the disclosures—or lack thereof—from ISPs.

Even the strongest network neutrality rules had carve-outs for "reasonable network management practices." For example, an ISP could prioritize voice and gaming traffic over file-sharing traffic if it determined that doing so would improve the efficiency of the network or the service that its subscribers would receive. Programmable networks enable a much richer set of management practices involving differential treatment of network traffic—including prioritization of latency-sensitive traffic or the blocking of attack traffic—many of which could be construed as "reasonable network management practices". For example, SDX allows networks to make decisions about interconnection, routing, and prioritization based on specific applications, which creates new traffic management capabilities that raise interesting questions in the context of net neutrality. Which of these new capabilities would constitute an exception for "reasonable network management practices,"[21] and which might be viewed as discriminatory?

Additionally, the automation of network management may make it increasingly difficult for operators to figure out what is going on (or why?). Some forwarding decisions may be more difficult to understand or explain if they are driven by a complex feature set and fully automated. Determining what "transparency" even means in the context of a fully automated network is a rich area for exploration at the intersection of network technology and telecommunications policy.[22] Even concepts seemingly as simple as "no blocking"—which state that an ISP should not block any type of traffic—have some nuance for network management and security, when considering that an ISP would need to block attack traffic. Yet, most decisions to block attack traffic today are deliberate and *post hoc*. These types of decisions (and investigations) may become more complicated in the context of *automatic* mitigation of attack traffic, implementation of takedown requests, or enforcement of copyright. Does every single traffic flow that is blocked by a network intrusion detection system need to be disclosed? How can ISPs best disclose the decision-making process for each blocking decision, particularly when either: (1) the algorithm or set of features may be difficult to explain or understand or (2) doing so might aid those who aim to circumvent these network defenses?

Programmability also enables fine-grained customization of network behavior, based on application, user, and device.

[21] *Id.* at 131 (defining reasonable network management practice as "appropriate and tailored to achieving a legitimate network management purpose, taking into account the particular network architecture and technology of the broadband Internet access service").

[22] *See also id.* at 126–29 (providing a history of the transparency rule).

Technology that enables fine-grained decisions about network traffic forwarding could naturally result in circumstances where every user has a different network experience. A future network might, for example, determine that a user prefers gaming to video streaming or file sharing and subsequently prioritize traffic flows differently than for a user who routinely shares files. Such a practice might even be characterized as network management. Disclosure of these practices could also become increasingly complex, particularly if some of the customization algorithms are based on machine learning and statistics that automatically prioritize certain flows without a human in the loop.

C. Jurisdictional Borders and Nation-State Concerns

Programmable networks and improved network measurement and inference capabilities may make it increasingly likely that network operators may be able to route traffic along network paths that traverse—or avoid—specific countries or regions. Improved geolocation of network infrastructure and the ability to configure fine-grained routing behaviors increasingly enable this type of functionality. The ability to configure network infrastructure to ensure that certain traffic flows avoid certain regions may have broader geopolitical implications for the Internet.

Consider, for example, the recently adopted net neutrality legislation in California.[23] The ability for a state to impose regulations on Internet traffic presumes the ability for an ISP to identify which traffic flows are contained within the state and which network traffic flows are interstate (or international). On the one hand, given the ability to automate flexible routing policies, an ISP might be able to identify which traffic flows are intrastate and thus subject to state laws. On the other hand, an ISP could use programmable networking to *control* how traffic flows—for example, intentionally routing traffic across state or national borders to change the applicable law of the traffic. As another example, previous work has explored how Internet routing could be used to divert domestic traffic across borders, subjecting the traffic to different laws concerning surveillance and data collection.[24] Today, manipulating traffic in this fashion is fairly challenging because the Internet's routing protocol, the Border Gateway Protocol (BGP), allows only fairly coarse-grained manipulation of

[23] CAL. CIV. CODE §§ 3100-04 (2018). *But see* David Shepardson, *California Will Not Enforce State Net Neutrality Law Pending Appeal*, REUTERS (Oct. 26, 2018, 12:33 PM), https://www.reuters.com/article/us-usa-internet/california-will-not-enforce-state-net-neutrality-law-pending-appeal-idUSKCN1N02KU [https://perma.cc/BY4B-HZ4S].

[24] *See* KRISTIN M. FINKLEA, CONG. RESEARCH SERV., THE INTERPLAY OF BORDERS, TURF, CYBERSPACE, AND JURISDICTION: ISSUES CONFRONTING U.S. LAW ENFORCEMENT 16–17 (2013).

network traffic.[25] Programmable networking, however, could make such manipulation easier to implement, perhaps even allowing a network to divert a single traffic flow, after associating that flow with an application, device, or user.

D. Liability and Failures

Network infrastructure experiences continuous faults, failures, and misconfigurations.[26] Typically, when a network experiences these types of disruptions, a network operator is tasked with diagnosing and correcting the problem. Programmability and the accompanying automation may ultimately make it more difficult for a network operator to determine the cause of poor performance, failure, misconfiguration, or other disruption, particularly if the algorithms that make decisions about traffic forwarding in the network make decisions that are difficult for a network operator to explain.

In particular, the rise of deep learning algorithms that make decisions based on non-linear models can often make it challenging to determine the particular factors that may have resulted in a configuration change or other change in network behavior. In conventional networks, a failure or performance degradation can often be traced to a particular underlying cause or specific configuration change—and reverted, if possible. If forwarding decisions are made based on complex, non-linear models that incorporate a wide variety of inputs, then decisions may sometimes cause unexpected behaviors that are difficult for humans to reason about, and even more challenging for a human operator to correct if it is difficult for the human to control or override automated decision-making. Operator liability is already an open question in other areas of automation, such as the context of autonomous vehicles.[27] Questions of liability for network failure are certain to arise in the future, as well.

In some cases, automated control may make it more difficult to attribute fault or liability to poor performance, which may make it more difficult to hold individual parties accountable for everything from contract or SLA violations to disclosures about performance, prioritization, and general network management practices.

[25] *See* Kevin Benton & L. Jean Camp, Examining the Jurisdictions of Internet Routes to Prevent Data Exfiltration 2–3 (Oct., 2016) (unpublished manuscript), https://ssrn.com/abstract=2753133 [https://perma.cc/QJ7R-JLBA].

[26] *See* Daniel Turner et al., *California Fault Lines: Understanding the Causes and Impact of Network Failures*, 40 ACM SIGCOMM COMPUTER COMM. REV., Oct. 2010, at 315, 316.

[27] RJ Vogt, *GM Settles First-Known Suit over Self-Driving Car Crash*, LAW360 (June 1, 2018, 10:56 PM), https://www.law360.com/articles/1049776 [https://perma.cc/A56R-JJEB].

E. *Privacy*

Disclosures about data collection and use are generally challenging for consumers (and even technical experts) to read. They can be vague, overly broad, out-of-date, and generally a poor reflection of technical practice. While a large aspect of the shortcoming of privacy disclosures amounts to the limitations of various legal frameworks, some of the problems have been exacerbated by the inability to perform fine-grained network measurement. In particular, legacy network measurement tools are generally either too fine-grained to gather at scale or too coarse-grained to yield helpful information about the performance of individual applications.

Because of the coarse-grained methods for data collection, network operators sometimes collect more data than they need for a particular task, even though the data itself might still be at an inadequate granularity for performing the desired task. One example of such data is IPFIX (or NetFlow) data, which gathers metadata about individual traffic flows in the network.[28] Although this mechanism records some information about each flow, such as the time of the flow, the source and destination IP addresses, and the number of bytes and packets in the flow, it contains no information about (1) packet loss or timings or (2) the actual service that corresponds to the flow (e.g., YouTube video). While the application or service can sometimes be directly inferred, this type of data is still wholly inadequate for capturing critical performance information such as application quality, let alone user experience. On the other hand, networks can gather so-called packet traces, which are essentially full recordings of the traffic as it passes through the network; in such a scenario, everything in the traffic is visible, subject to end-to-end encryption. Neither of these extreme design points is particularly useful at “finding the needle in the haystack,” or identifying precisely the network traffic data that an operator needs to address the network performance or security task in question.

The rise of programmable networking—and in particular, programmable network measurement—creates some opportunities in this regard. The ability to more precisely and accurately specify traffic flows of interest creates the potential to capture *only the traffic that is needed* to perform the specific task at hand. In the case of some queries, coarse-grained network traffic statistics may be sufficient; in other cases, perhaps queries from the Domain Name System (DNS) would help useful, without requiring the need

[28] *See* Rick Hofstede et al., *Flow Monitoring Explained: From Packet Capture to Data Analysis with NetFlow and IPFIX*, 16 IEEE COMM. SURVEYS & TUTORIALS 2037, 2037–39 (2014).

for a full packet capture. The rise of new programmable network telemetry systems such as Sonata makes it possible to tailor the data that is gathered to the query or task that the network operator aims to achieve, thus substantially reducing the risk that data is over-collected.[29] It is also worth noting that, in this case, user privacy also aligns with scale and the ability to capture network traffic at high-speed. Specifically, capturing all traffic as it traverses the network (deep packet inspection, or DPI) not only poses privacy risks to users but is also more difficult to perform as traffic rates increase. Programmable network telemetry systems should be a boon for both system scalability and user privacy.

III. What About Network Virtualization?

Network virtualization is often discussed in the same context as SDN—sometimes discussions even conflate the two topics—and so it is worth explaining the relationship of network virtualization to SDN. Network virtualization is, in fact, a distinct topic from SDN. On the one hand, SDN separates network "control plane" software from "data plane" routers and devices; virtualization, on the other hand, involves creating virtual server and network topologies on a shared underlying physical network. In short, SDN is a technology that facilitates network virtualization, but the two are distinct technologies.

Nonetheless, network virtualization presents many timely issues at the intersection of technology and policy in its own right. For one, the rise of Infrastructure as a Service (IaaS) providers, such as Amazon Web Services, as well as other multi-tenant data centers, introduces questions such as how to enforce SLAs when isolation is imperfect as well as how IaaS providers can be stewards of potentially private data that may be subject to takedown requests, subpoenas, and other actions by law enforcement and other third parties. The forthcoming Supreme Court case *United States v. Microsoft Corp.*[30] concerns law enforcement access to data stored abroad. Does the data actually live overseas, or is this merely a side effect of global, virtualized data centers? Virtualization presents a variety of other interesting questions at the intersection of technology and policy, as well, as it enables content to be hosted almost anywhere in the network—and quickly moved from one place in the network to another. A significant and growing fraction of Internet traffic is exchanged with distributed cloud services, which largely rely on virtualization technologies. These technologies have dramatically shifted Internet traffic patterns in ways that are fast reshaping the economic and regulatory

[29] *See* Arpit Gupta et al., *supra* note 11.

[30] 138 S. Ct. 1186 (2018).

landscape in areas such as Internet interconnection. These topics and questions are extensive enough to warrant their own paper, since most of them are distinct from the questions discussed here.

CONCLUSION

The capabilities of SDN—and, more generally, programmable networks—introduces new questions the intersection of policy, law, and technology, centering on the implications of more flexible monitoring and more automated decision making. Programmable network monitoring on the one hand could bring a new golden age for Internet transparency, as new tools make it possible for operators, users, and policymakers to ask questions about the operation of the network that were previously technically untenable. The same flexibility could pose new threats to privacy, if the technologies are not carefully and thoughtfully designed, but at the same time this programmability could enable in-network computation and aggregation that could in fact *improve* user privacy by performing more computation at the edge, avoiding the need to collect and warehouse large volumes of data in centralized repositories.

The automation enabled by programmable networking also brings with it its own set of challenges and opportunities. On one hand, automation could improve the performance of networks, reducing downtime and enabling more sophisticated service offerings with improved network efficiency. On the other, automation can, in some cases, make it more challenging to explain the causes of a particular decision about network operations, which could introduce new questions concerning both troubleshooting and liability.

In summary, network programmability is ultimately a double-edged sword for tech policy; the outcomes will ultimately depend on engineers and policy-makers carefully considering how programmability affects a wide range of concerns, including those raised in this paper.