Research Statement

Oliver Michel

February 2021

I design and build systems to make future networks more reliable, more secure, and easier to manage. My focus is on network management and the trend toward autonomous, closed-loop control which requires gathering and acting upon fine-grained insight into the network. Doing so without impacting the network's operation or incurring disproportionate cost is the goal of my work. Instead of repeatedly finding answers for specific sub-problems, I aim for universal solutions by carefully designing systems using a holistic, end-to-end approach across platforms and technologies, drawing techniques from computer and systems architecture, performance optimization, and network programmability.

1 Research Area and Approach

Today's networked applications are ubiquitous and require reliable, secure, and high-performance network infrastructure. The significance of these applications continues to grow, spanning domains from communication to entertainment and healthcare. As a result, traffic rates in wide area and data center networks are, more often than not, on the order of several Terabits and hundreds of millions of packets per second. Yet, it is not only the volume of traffic carried in modern networks which is growing rapidly; the frequency and magnitude of attacks, as well as the complexity of network infrastructures and operations, are higher than ever before. And there is no end in sight. As the number of connected devices increases rapidly and more applications adopt cloud computing models, scaling network infrastructures while closely monitoring their operation and quickly reacting to attacks or changes in demand will continue to be a major challenge for the years to come.

My research aims at making networks more dependable and secure. I am guided by the broader vision of simplifying network management and enabling large and complex networks to run autonomously, driven by fine-grained insight into their operation. Toward this goal of closed-loop network control, my particular interest lies in integrating components across control and data planes as well as hardware and software platforms, carefully partitioning workloads, finding the right abstractions for interfaces between components, and optimizing the end-to-end system to be practically deployable. Over the past years, we have seen work enabling parts of this aspiration with solutions based on in-network data structures, machine learning approaches, and streamlined software packet processors, to name a few. Yet, we are still lacking a common abstraction and higher layer to systematically orchestrate and configure this heterogenous set of technologies for specific network management requirements. These requirements and objectives might extend into the application layer to be, for example, on the granularity of remote procedure calls in a micro-service environment requiring coordination and correlation between the network and application domains.

My prior work and background in most of the before-mentioned areas are integral to understanding the complex issues at hand and designing end-to-end network management systems. This work has focused on hardware-accelerated network telemetry [8], high-performance network analytics [4, 5, 6], machine learning-based intrusion detection [1], and abstractions for network control [2, 3, 7]. All works have been driven by a holistic approach questioning previous assumptions and beliefs. For example, we have demonstrated that software is not inherently incapable of providing high performance when carefully optimized for specific workloads. Lastly, through experience in industry, I have learned how to build and deliver large-scale production software systems. I apply the same methods to my research projects with the ultimate goal of sharing them with the community.

2 Prior Research Work

During my undergraduate and graduate studies I worked on various topics related to software-defined networking, focusing on improving performance of networked applications through SDN, as well as the architectures and abstractions of SDN systems themselves. Later, during the second half of my Ph.D. program, my focus shifted to network telemetry and scalable network analytics leveraging insights from my earlier works.

2.1 Abstractions and Performance in Software-Defined Networks

Policy Routing on OS-Level Identifiers Network devices generally rely on layer 2 to layer 4 identifiers to make forwarding decisions without a notion of the exact origin of a packet in terms of the sending user or process. In our work on fine-grained policy routing, we enable OS-level information, like user or process identifiers or the fingerprint of an executable, to be used in the data plane. Our system includes a shim layer that tags packets based on such identifiers and a programmable switch parsing tags and taking forwarding decisions (e.g., drop or redirect to an IDS) based on these tags and network-wide security policies [2].

SDN Controller Design Network controllers for SDN are often implemented as monolithic applications through a library or composition abstraction. In our work on SDN controllers we extended Linux and leveraged its APIs and software ecosystem to serve as a practical SDN controller. Applications for this controller run as separate processes, can be written in any language, and interact with the network through standard file I/O APIs; they further benefit from common and powerful technologies such as the virtual file system layer for distributed control and namespaces for control isolation [7].

Adaptive Source Routing Source-controlled routing coupled with path performance estimation is a promising approach to improve quality of service in wide-area networks. In this work, we studied the efficacy of source routing together with active probing to reduce latency and packet loss for latency-sensitive applications. Selective packet replication along multiple paths can further reduce tail latency while keeping the bandwidth overhead small [9]. We also conducted a case study for the field of preclinical medical care where an ambulance is connected to a hospital team via multiple mobile carriers.

Cloud Resource Fragmentation Mapping and repeatedly scaling virtual networks in cloud environments introduces fragmentation in the substrate network (akin to fragmentation on a storage device). Instead of further improving embedding algorithms to tackle this problem, we proposed new management primitives leveraging network migration techniques to effectively defragment the network. Simulations showed that our techniques significantly improve network performance while maintaining high utilization of the infrastructure, thus increasing provider revenue [3].

2.2 Scalable Network Telemetry and Analytics

My prior work on network telemetry and analytics has focused on the question of whether it is possible to export and perform meaningful analytics on every packet in cloud-scale networks without sacrificing performance. By taking a holistic, systems-focused approach, we were able to show that this goal is indeed within reach.

Lossless Line-Rate Packet Telemetry Our first line of work in this area explored whether telemetry systems can harness the performance of programmable line-rate switches while also meeting requirements for efficient integration with analytics platforms and practical deployment. Toward answering this question, we introduced *Flow [8], a practical hardware-accelerated telemetry system that is not only efficient, but also supports concurrent measurement and dynamic queries through carefully partitioning between hardware and software components. *Flow's design plays to the strengths of both programmable switches and general purpose hardware. Instead of compiling entire queries to a switch, *Flow places parts of the select and grouping logic that is common to all queries into a match+action pipeline in the data plane. The pipeline operates at line-rate and exports a stream of records.

Flexible Network Analytics in Software While being a suitable platform for feature extraction and basic grouping, the constraints of hardware in both available resources and types of computation supported fundamentally limit the ability to perform a wider range of analytics tasks (e.g., machine learning) in the data plane. Previous software-based approaches to network analytics rely on pervasive filtering, sampling, or

early feature aggregation to cope with high traffic rates in networks, limiting possible applications and their accuracy. In our work on software analytics, we demonstrated that high performance network analytics can be realized without these sacrifices. The core idea is to leverage programmable switches to make software analytics better, rather than accelerating select measurement applications by compiling them to hardware. We eliminated software choke points by pushing partitioning and fixed preprocessing into hardware. Analytics tasks are then performed in highly parallel software pipelines providing virtually unlimited programmability and flexibility. The resulting system, Jetstream [4, 5], is a hardware-software co-designed network analytics system that supports any analytics task entirely in software while also being able to process up to 250 million packets per second on a 16-core commodity server for different common analytics tasks, a performance improvement of one to two orders of magnitude compared to previous systems.

Retrospective Queries on Network Records Finally, in our work on persistent and interactive queries on network records (PIQ) [6], we took first steps toward a persistence system that supports live queries and retrospective queries for network debugging and auditing. We explored the requirements of such a system, identified time-series databases as a promising starting point, and designed strategies for using modern database engines with state-of-the-art network telemetry and analytics systems.

3 Ongoing and Future Research Directions

I plan to continue working toward more dependable and secure network systems focusing on improved management abstractions and practical deployment considerations. To achieve this goal, I will build upon the promising results of my prior work in telemetry and analytics. Moreover, I aim at expanding my scope and taking insights gained from prior work into other areas which may include radio area networks or higher-level abstractions, such as service meshes. There are two main areas in particular which I am currently exploring and intend to continue emphasizing in the near future.

3.1 Reducing the Footprint of Network Telemetry and Analytics Systems

Continuous, fine-grained network monitoring is imperative to ensure the correct and reliable operation of compute and network infrastructures. In comparison to, for example, provisioning more servers, however, allocating resources for monitoring only indirectly and on longer time scales results in increased earnings for a network or cloud operator. Therefore, going forward, I believe that improving the efficiency of monitoring systems as much as possible is the key to the widespread adoption of fine-grained monitoring and subsequently making tomorrow's networks more resilient, secure, and performant. There are two concrete directions that have the potential to achieve this goal.

First, I believe network telemetry systems can be designed to be more adaptive. Networks exhibit vastly different characteristics in terms of scale, traffic they carry, and the types of applications they support. Different setups require different sets and granularities of metrics, and different applications are prone to different types of attacks. Having the ability to define the *dials* exposed by the network through data plane programmability allows us to precisely tailor a telemetry system to a particular network setup. We can choose which traffic to analyze in full detail, which traffic to aggregate and compress early, and, taking into account the resource constraints of line-rate switches, which measurement tasks to run in the data plane. The challenge here is to make these choices systematically and in an automated fashion given the properties of the traffic, the monitoring objectives, and a budget in terms of switch, server, and network resources.

A second direction concerns even more fine-grained placement of individual system components that are part of an end-to-end network monitoring system, which spans all the way from the raw packet on a wire to a meaningful insight for an operator or network automation platform. Our previous work has focused on partitioning this pipeline between line-rate switches and general-purpose CPUs. In the future, I plan to explore even more specialized placement of functions, leveraging SmartNICs, FPGAs, and GPUs, finding the most suitable platform for a given part of the pipeline. In pursuit of this goal, we are currently exploring the applicability of eBPF programs that appear to be a powerful container for lightweight analytics computations. A single task packaged as an eBPF program can run on a SmartNIC, in the kernel, as well as in user space allowing for reusability and flexible placement strategies across platforms. In parallel, we are

also exploring vector instruction sets, such as AVX-512, to accelerate machine learning tasks in the context of network analytics where either a programmable switch or a SmartNIC efficiently pre-formats a stream of input features to be directly loaded into vector registers for efficient computation. While I plan to initially tackle these areas within the context of network monitoring, finding answers and better abstractions for fine-grained placement of system functions is important for a wide range of networked applications, including distributed machine learning or key-value stores.

3.2 Simplifying Data-Driven Network Management

Autonomous, data-driven network control has the potential to make networks more resilient and secure. The ability to collect network-wide measurements and analyze them efficiently is a key prerequisite for such closed-loop network control. Filling all the gaps in this control loop is imperative; yet, as alluded to in the first section of this research statement, we are still facing challenges in integrating the loop's components. I am interested in exploring the interfaces between measurement, telemetry, analytics, and automation components with the goal of finding a higher-level abstraction to define (and subsequently compile and deploy) network control loops. A first step toward this could be as simple as applying the widely-used and well-known match+action abstraction at a higher level, such that an operator matches on a coarse-grained condition (e.g., "SYN flood") that is associated with a network-wide management policy (e.g., "block origin").

A second direction here revolves around record persistence. On top of running live analytics on a stream of packets, a network operator might want to store traffic records or network metrics to answer questions about a past state of the network. This is important for use cases such as debugging previously experienced poor network performance, performing forensic analysis, or complying with legal obligations. In our prior work on persistent and interactive queries, we looked into time-series databases for this application. Even when applying various optimizations, however, data ingestion rates in existing systems are still orders of magnitude below network traffic rates. We are currently exploring how to build a lightweight but high-performance and space-efficient persistence layer that allows for basic queries over stored network records without having to maintain a full and complex database index.

References

[1] Greg Cusack, Oliver Michel, and Eric Keller. Machine learning-based detection of ransomware using SDN. In *Proc. 2018 ACM Workshop on Security in SDN and NFV*, SDN-NFV Sec., 2018.

[2] Oliver Michel and Eric Keller. Policy Routing using Process-Level Identifiers. In *Proc. 3rd IEEE Symposium on Software Defined Systems*, SDS, 2016.

[3] Oliver Michel, Eric Keller, and Fernando M.V. Ramos. Network defragmentation in virtualized data centers. In *Proc. 6th IEEE Conference on Software Defined Systems*, SDS, 2019.

[4] Oliver Michel, John Sonchack, Greg Cusack, Maziyar Nazari, Eric Keller, and Jonathan M. Smith. Software Packet-Level Network Analytics at Cloud Scale. *IEEE Transactions on Network and Service Management*, 18(1), 2021.

[5] Oliver Michel, John Sonchack, Eric Keller, and Jonathan M. Smith. Packet-level analytics in software without compromises. In *Proc. USENIX Workshop on Hot Topics in Cloud Computing*, HotCloud, 2018.

[6] Oliver Michel, John Sonchack, Eric Keller, and Jonathan M. Smith. PIQ: Persistent interactive queries for network security analytics. In *Proceedings of the ACM International Workshop on Security in Software Defined Networks and Network Function Virtualization*, SDN-NFV Sec., 2019.

[7] Matthew Monaco, Oliver Michel, and Eric Keller. Applying operating system principles to sdn controller design. In *Proc. 12th ACM Workshop on Hot Topics in Networks*, HotNets, 2013.

[8] John Sonchack, Oliver Michel, Adam J. Aviv, Eric Keller, and Jonathan M. Smith. Scaling Hardware Accelerated Network Monitoring to Concurrent and Dynamic Queries With *Flow. In *2018 USENIX Annual Technical Conference*, ATC, 2018.

[9] Ashish Vulimiri, Oliver Michel, P. Brighten Godfrey, and Scott Shenker. More is less: Reducing latency via redundancy. In *Proc. 11th ACM Workshop on Hot Topics in Networks*, HotNets, 2012.