Research Statement

Satadal Sengupta

I am a ***networking and systems researcher*** committed to improving the performance and security of networked applications as they face an unprecedented surge in scale, complexity, and resource demands. My vision is to make networks ***deeply observable in real time***, enabling immediate and automated control that enhances application performance and security. To realize this vision, I build techniques and systems that allow operators of large networks to: (i) ***measure*** key metrics such as delay, jitter, and video frame-rate from network traffic [1, 2], (ii) ***analyze*** these measurements to detect performance and security problems, such as increased delay during a route hijack [2, 3], and (iii) ***control*** network behavior to mitigate such problems, such as by dropping, rate-limiting, or rerouting traffic [3, 4]. These capabilities run in ***real time at line rate*** on high-speed programmable hardware, such as programmable switches and smartNICs, despite their stringent memory and compute constraints. Collectively, my work enables ***closed-loop control*** for modern networked applications, laying the foundation for ***self-driving networks***. The artifacts of my work are open source and are being leveraged by academia and industry to improve the status quo of network monitoring and control.

Why do networks need deep observability? In 2025, the average Internet user consumes over 4 GB of data per day through increasingly sophisticated applications such as live streaming, video conferencing, online gaming, and AI chatbots—reflecting the unprecedented scale and complexity of today's Internet. Yet, despite decades of innovation, the underlying infrastructure does not guarantee performance or security and remains prone to packet loss, excessive latency, and suboptimal routing. To compensate, the Internet relies on a diverse set of sophisticated protocols that aim to provide reliable and secure communication over a fundamentally best-effort substrate. These protocols span the entire stack—from application-layer systems like the Domain Name System (DNS) to physical-layer technologies such as Wi-Fi. They also range from open-source, configurable protocols like the Transmission Control Protocol (TCP) to proprietary and opaque implementations such as Zoom's variant of the Real-Time Communication (RTC) protocol. However, despite these mechanisms, severe performance and security degradations persist. To address such issues, operators of large networks (e.g., enterprise or cloud) have traditionally relied on coarse-grained monitoring, either at the flow level or at the packet level for a small sample of flows. However, many of the most damaging degradations stem from problems that are either ***short-lived*** or ***stealthy***, yet affect the network's performance and security disproportionately. These are precisely the kinds of problems that coarse-grained techniques are least equipped to detect. As a result, operators have blind spots with serious consequences. For example, a survey of enterprise users of video-conferencing applications shows that one in four users experiences video lags during peak hours, often caused by transient congestion events. Meanwhile, cryptocurrency users lose millions of dollars every year to stealthy routing attacks. These examples reveal a mismatch between significant network issues and existing troubleshooting tools. To address this mismatch effectively, the monitoring and control available to network operators must be both ***real-time*** and ***fine-grained***, which is enormously challenging at Internet scale.

Opportunity, challenge, and research approach. Fortunately, state-of-the-art networking hardware platforms, such as programmable switches and smartNICs, combine production-scale speed (Tbps on switches, hundreds of Gbps on smartNICs) with programmability, creating a new opportunity. However, this opportunity comes with important caveats. To sustain line-rate speeds, their programmability is extremely limited: for example, the Intel Tofino switch severely constrains memory, limits computation stages, forbids loops, and disallows complex arithmetic. Meanwhile, the network protocols that govern modern traffic, such as TCP and RTC, are inherently complex, involving multiple, semantically distinct packet types, variable packet boundaries, and sophisticated mechanisms to recover from loss and reordering. Furthermore, tracking and analyzing traffic governed by these protocols requires cross-packet state over a variable number of packets. As a result, naive solutions that do not strike the right balance between hardware resources and algorithmic sophistication either do not fit, rapidly exhaust memory, exceed computation stages, or incur too many false positives while detecting problems. My research walks the ***tightrope between logical complexity and resource scarcity*** by carefully dissecting applications and their underlying protocols to identify hardware-amenable approximate solutions.

Research Contributions. Guided by this approach, my research has addressed four distinct yet complementary problem domains. First, I ***demystified the network behavior*** of popular applications and network protocols—

including proprietary applications such as YouTube and Zoom, and custom implementations of open-source protocols such as Transport Layer Security (TLS) [2, 5–7]. Second, my deep understanding of these applications enabled me to design techniques and systems for ***continuous monitoring*** of their key performance and security indicators, such as round-trip time (RTT) and video frame-rate, directly inside the network [1, 2]. Third, I designed systems to ***analyze*** these indicators in real time and ***detect anomalies***, for example, by performing changepoint detection on an RTT time-series, to detect attacks and performance issues as soon as they occur [1, 3, 8, 9]. Finally, I leveraged the insights and experience gained from these studies to identify and remedy deficiencies in the ***infrastructure*** supporting these applications, thereby improving their scalability and quality of experience (QoE) [4, 8, 9]. Below, I elaborate on our work in each problem domain.

1 Demystifying Proprietary Applications and Protocols

The deployment and operation of every major Internet application is a large collaborative effort involving application developers, web or mobile developers, cloud providers, Internet service providers, enterprise or campus network operators, etc. As a result, an application's performance, security, and availability depend on the competence and coordination of many such independent stakeholders. Yet the individual components of these applications are often opaque for business reasons, making effective collaboration difficult. For example, network operators can provision bandwidth, mitigate congestion, and strengthen security at the edge, but they often lack insight into how applications like Zoom or YouTube behave, limiting their ability to intervene in ways that improve user experience. This black-box nature also raises privacy and regulatory concerns, since neither users nor regulators have the visibility needed to assess whether performance and security align with expectations or policy. To address these issues, a significant part of my research focuses on uncovering how popular Internet applications and protocols—especially proprietary systems with opaque behavior—operate from the network's point of view. This work spans a wide range of traffic types, including interactive video conferencing [2], adaptive video streaming [6], encrypted mobile traffic [5], and affordable Internet access services [7, 10], demonstrating how visibility can be improved across protocol layers and application domains.

1.1 Demystifying Zoom

Video-conferencing applications such as Zoom are now critical to business, healthcare, and education, yet their internal behavior is largely opaque to network operators. This motivated our effort to extract ***per–media stream performance metrics*** from passive network traffic already flowing between real users, without relying on end-host instrumentation [2]. Our primary challenge was that Zoom's RTC protocol is proprietary, undocumented, and mostly encrypted, making it unclear which portions of each packet—if any—exhibit meaningful structure. To address this, we performed a detailed ***entropy analysis*** of packet bytes to identify stable, low-entropy regions that Zoom leaves unencrypted. These previously undocumented fields revealed consistent offsets and bit patterns that enabled us to recover sequence numbers, timestamps, frame identifiers, and media-type markers. Using these fields, combined with controlled experiments to validate their semantics, we reconstructed Zoom's media hierarchy: grouping packets into frames, frames into streams, and streams into meetings while distinguishing among audio, video, and screen-share traffic. This reconstruction allowed us to compute bitrate, frame rate, jitter, and end-to-end latency directly from encrypted packet traces. Applying these techniques to a month-long campus dataset revealed a long tail of poor performance across key QoE indicators like jitter and bitrate, offering the first large-scale, network-centric view of Zoom's behavior.

1.2 Pre-Ph.D. Work on YouTube, TLS, and Free Basics

Uncovering YouTube's Adaptive Streaming Behavior. YouTube accounts for a substantial share of global Internet traffic, yet its adaptive bitrate (ABR) logic is proprietary and opaque, making it difficult to understand how the client maintains quality under fluctuating network conditions. Through browser instrumentation and controlled experiments, we correlated network conditions with adaptation decisions and uncovered a key behavior missed by prior work: YouTube adapts both the bitrate and the segment length of video segments, rather than bitrate alone [6]. This previously undocumented mechanism clarified why earlier studies significantly overestimated YouTube's data waste: when the network suddenly improves, ABR algorithms often switch to

a higher-quality version of the same soon-to-be-played segments already in the buffer, leading to redundant downloads. YouTube's simultaneous adaptation of segment length greatly reduces this waste.

Exploiting TLS Diversity to Classify Encrypted Mobile Traffic. The widespread adoption of TLS protects user privacy but complicates traffic classification (TC), a critical step for service differentiation. By analyzing TLS-encrypted traffic from 175 mobile apps, we showed that inconsistencies in TLS configurations and diversity in cipher suites introduce distinctive patterns in encrypted payloads [5]. By combining these payload-derived features with traditional packet-level statistics in a Random Forest classifier, we improved classification accuracy from 62% to 95%. This work demonstrated that even encrypted traffic can reveal meaningful application signatures when protocol implementations are carefully analyzed.

Understanding Zero-Rated Internet Access. Zero-rated services such as Meta's Free Basics aim to lower the barrier to Internet access, yet their real-world scope and performance remain opaque. Using a crowdsourced measurement platform across 15 countries and our own Free Basics-eligible web services, we evaluated both reachability and performance [7, 10]. We found that many popular sites were excluded, nearly 60% of homepage links pointed outside the Free Basics' ecosystem, and free-service pages were four times slower than paid alternatives. At the same time, usage data showed substantial global reach and clear demand, highlighting how zero-rated services can expand access while simultaneously constraining performance and content diversity.

2 Enabling Continuous In-Network Monitoring

While the first thread of my work focuses on understanding how modern applications and protocols behave, the second focuses on how network devices can ***continuously observe*** this behavior in real time. Many popular measurement tools rely on ***active*** probes, but they fail to capture application-specific RTTs, introduce additional traffic load, and may be blocked by remote hosts or networks; in contrast, ***passive*** measurement of traffic already flowing through the network provides more accurate and representative estimates. In this thread, I develop algorithms and systems that harness high-speed programmable network hardware to create real-time measurement engines operating directly on live traffic, even under severe memory and compute constraints.

2.1 Continuous Round-Trip Time Monitoring

RTT is a critical network metric: it correlates with application QoE, helps expose routing attacks, and is central to minimizing delay in low-latency communication. Existing passive RTT measurement techniques only consider packets during connection establishment, which provides a misleading view for long-lived flows that evolve over time. Programmable switches offer an opportunity to measure RTTs continuously and in real time by recording outgoing packets in a hash table and matching them with returning acknowledgments. However, doing this correctly and at line rate introduces two key challenges. The first is correctness: TCP ensures reliable delivery of traffic using mechanisms such as *retransmission* and *reordering*, which can cause a naive packet-matching approach to associate the wrong packets and acknowledgments, resulting in artificially inflated RTT values. The second challenge is memory efficiency: TCP sends acknowledgments selectively for efficiency, meaning that many outgoing packets never receive a direct match and accumulate, quickly exhausting the limited memory available on programmable hardware. While timeouts might appear to solve this, short timeouts bias the system against large RTTs, whereas long timeouts fail to reclaim memory quickly enough, making them ineffective in practice. To address these issues, we developed ***DART*** (Data-plane Actionable Round-trip Times) [1]. Rather than relying on a single hash table for packet matching, DART ensures correctness by maintaining an auxiliary table that tracks the ***valid range of sequence numbers*** for each flow, allowing the system to detect retransmissions and reordering and discard distorted RTT samples. DART achieves memory efficiency through ***lazy eviction***: when a hash collision occurs, outdated packet records that fall outside the valid range are removed to make space for new records. Our P4 prototype on the Intel Tofino switch recovered 99% of the RTT samples produced by an offline, unlimited-memory baseline software tool called ***tcptrace***, while sustaining production speeds. DART demonstrates the feasibility of continuous RTT monitoring for timely control.

2.2 Enabling Passive Measurement of Zoom Performance

The structural understanding obtained from our earlier Zoom analysis also enabled us to design mechanisms for ***lightweight, stream-level monitoring*** that rely only on the unencrypted header fields available at line

rate [2]. Unlike host-based telemetry or payload inspection approaches, our goal was to determine what could be computed efficiently inside the network, using only the fields exposed by Zoom's wire-format. Building on the media-hierarchy reconstruction from our earlier analysis, we identified a minimal subset of header fields that permit the classification of Zoom packets into audio, video, and screen-share streams and allow the computation of bitrate and inter-arrival jitter without parsing or decrypting payloads. We evaluated this approach on a 12 hour-long campus trace covering 1.8 billion Zoom packets and 59,020 media streams, showing that these metrics can be derived accurately from packet headers alone and with modest computational overhead. These results demonstrate that continuous, application-aware monitoring of real-time conferencing traffic is feasible in the data plane, making it a practical fit for programmable switches and smartNICs operating at production speeds.

3 Automating Analysis and Control

The power of continuous, fine-grained monitoring is fully realized when the network is able to interpret these measurements and act on them immediately. Many disruptive network problems, including transient congestion resulting in sudden QoE drops, and stealthy long-distance routing attacks, cause damage before operators have a chance to intervene manually. Their impact can be mitigated only if the analysis and control loop operates at the same granularity and speed at which measurements are taken. This motivates the third thread of my research, which develops systems that automatically analyze real-time data-plane signals to detect anomalies and trigger control actions, enabling networks to respond to emerging issues within milliseconds.

3.1 Mitigating Long-Distance Routing Attacks

Despite major advancement in defenses over the last decade, route hijacks continue to pose a serious threat, particularly when attackers reroute domestic traffic through foreign jurisdictions, exposing users' data to different privacy and surveillance regimes without their knowledge. Existing defenses fall short: proactive mechanisms that enhance the security of routing protocols face well-known deployment barriers or limited attack coverage, while reactive control-plane monitors can miss stealthy, targeted hijacks. This motivated us to explore passively monitored ***propagation delay*** as an alternative detection signal [3]. Long-distance routing attacks necessarily route traffic over thousands of additional kilometers, often increasing RTTs by tens of milliseconds—an effect that cannot be concealed from the victim and manifests immediately when the attack begins. However, transforming delay into a practical detection signal raises two core challenges. The first is ***location-dependent detectability***, since the expected RTT increase varies with the geolocations of the sender, receiver, and attacker. Our system ***HiDe*** (Hijack Defense) addresses this by computing a per-prefix minimum possible RTT—the smallest physically realizable RTT if traffic were rerouted through a given threat region. The second challenge is the ***scalability*** of a real-time defense, as RTTs are noisy and computing and denoising at line rate is difficult. HiDe leverages DART to obtain accurate per-flow RTTs in the data plane and aggregates them into per-prefix minimum RTTs, which approximate propagation delay and filter out noise from congestion and the access network. HiDe then applies a switch-native changepoint detector, raising an alert and optionally blocking traffic when the measured minimum RTT exceeds the prefix-specific threshold. This captures the sharp, prefix-wide delay surge that accompanies long-distance routing attacks, while fitting entirely within the memory and computation limits of a programmable switch. HiDe detected all ethically launched real-world hijacks within 0.5 seconds, demonstrating zero false negatives by design. On 19 billion packets of campus traffic, it maintained a false-positive rate of below 0.012% and automatically corrected false alerts within a median of 0.75 seconds. In a nutshell, HiDe demonstrates the power of in-network analysis and control.

3.2 Pre-Ph.D. Work on Service Differentiation of Mobile Video Traffic

As mobile video traffic proliferates, streaming and interactive video flows increasingly compete for constrained bandwidth. ***MoViDiff*** enables network operators to differentiate these flows at mobile gateways and apply bandwidth policies accordingly [9]. We show that classifiers based on static IP-port combinations and payload inspection perform poorly, while ***packet-size distribution*** provides a reliable signal: streaming flows exhibit a long tail of large packets, whereas interactive flows use predominantly smaller packets. Leveraging this insight, MoViDiff uses a lightweight Support Vector Machine classifier to distinguish flow types with 96% accuracy and enforce real-time bandwidth allocation, improving interactive video QoE when bandwidth is limited.

4 Network-Boosted Applications

Finally, my research examines how the network can move beyond serving as a passive transport medium for application traffic and become an ***active contributor*** to application performance and scalability. I pursue this through two complementary strategies. First, I use programmable data planes to offload the most frequent and latency-sensitive components of an application's workload, enabling a substantial increase in performance and scalability. Second, I design network-aware application logic that combines real-time knowledge of network conditions with externally provided user preferences to improve QoE. Together, these approaches demonstrate how tightly integrating applications with network capabilities can significantly enhance application performance.

4.1 Scalable Video Conferencing using Principles of Software-Defined Networking

Selective Forwarding Units (SFUs) are central to video-conferencing systems: they process large volumes of media traffic under tight latency constraints, and their workload grows dynamically as participants join, leave, or toggle audio/video/screenshare. Moreover, SFU load scales quadratically with meeting size, since each additional participant increases the replication and forwarding work of the SFU for every other participant. Our earlier work on Zoom and follow-up analysis of the open-source SFU MediaSoup highlighted a key insight: the core functions of an SFU closely resemble ***traditional networking tasks***. Media replication maps naturally to ***multicast***, and selective forwarding to adapt media quality aligns with ***firewall-like filtering***. Building on this, we designed ***Scallop***, a hardware–software co-designed SFU inspired by SDN principles [4]. A high-speed data plane performs the high-frequency latency-sensitive operations—selective replication and packet forwarding—while a lean software control plane handles infrequent tasks such as connection initiation and management, and rate control. Designing Scallop required addressing three main challenges. First, we had to determine the right control/data-plane split, since SFUs are currently written as monolithic software. Second, enabling scalable packet replication demanded repurposing the data plane's multicast primitives, which are not designed to natively support the dynamic, meeting-specific replication required by SFUs. Third, Scallop needed to remain fully compatible with unmodified WebRTC clients, since WebRTC is the de-facto standard for video conferencing. This required lightweight sequence-number rewriting and careful handling of feedback packets. Implemented on a Tofino switch with a minimal software control plane, Scallop supports 10–200× more meetings for the same cost and incurs 26× lower forwarding latency compared to a 32-core software SFU, while maintaining compatibility with WebRTC. We also implement Scallop's data plane on the NVIDIA BlueField3 smartNIC, demonstrating the generalizability of its design.

4.2 Pre-Ph.D. Work on Hotspot-Aware Adaptive Video Streaming

Dynamic Adaptive Streaming over HTTP (DASH) selects per-segment video quality based on network conditions, but existing ABR algorithms assumed all segments are equally important, ignoring user-preferred scenes called ***hotspots***. ***HotDASH*** addresses this by opportunistically prefetching hotspot segments when bandwidth permits, while still optimizing bitrate, smooth playback, and low rebuffering akin to prior ABR algorithms [8]. In doing so, we face challenges in supporting out-of-order downloads, deciding when to prefetch under volatile bandwidth, and balancing hotspot quality with the other objectives. HotDASH resolves these using a custom prefetch-enabled DASH client and a deep reinforcement learning controller that makes prefetch and bitrate decisions jointly, achieving 16% higher QoE and 14% higher bitrate on average over state-of-the-art baselines.

5 Future Research Goals

I am eager to continue research driven by the evolving challenges faced by modern networked applications. In the short term, I plan to extend my work on closed-loop control to emerging classes of traffic, such as AI/ML workloads, which impose distinctive performance requirements. In the longer term, I also aim to advance the per-packet extraction of critical network metrics, thereby enriching the set of network features available to machine learning-based inference and control systems—an essential step toward realizing self-driving networks. Achieving these goals will require not only advances in networking research, but also engagement with network operators, industry partners, and broader communities that rely on performant and secure networked systems.

5.1 Research Directions

Comprehensive Closed-Loop Control. A natural next step in my research is to broaden the scope of closed-loop control to ***end-to-end closed-loop systems*** that span applications, protocol layers, and heterogeneous hardware. My work on demystifying proprietary applications has shown the value of deeply understanding how real applications behave in the wild—whether Zoom media traffic, YouTube's adaptation logic, TLS implementations, or zero-rated services. I plan to extend this line of inquiry to emerging application domains such as AR/VR, cloud gaming, and large-scale AI chat interfaces, whose traffic patterns, latency characteristics, and failure modes are still poorly understood. At the same time, the deep observability techniques I have built, from passive media-level metrics to continuous, per-packet RTT measurement, must expand beyond the network and transport layers to include ***cross-layer visibility***, reaching downward into cellular and Wi-Fi, and upward into the application layer. Finally, realizing truly immediate and automated control will require generalizing my in-network analysis and mitigation techniques beyond high-speed switches to a ***heterogeneous ecosystem of programmable platforms***, including smartNICs, FPGAs, and eBPF-enabled hosts.

Defense In Depth. A second direction I aim to pursue is the design of ***defense-in-depth systems*** that automatically distribute security functionality across heterogeneous programmable devices—including high-speed switches, smartNICs, eBPF-enabled hosts, and servers. These platforms differ widely in throughput, programmability, and resource budgets, making manual placement of defenses brittle and inefficient. The opportunity is to leverage these differences: switches can provide fast, approximate filtering; smartNICs can apply richer, stateful checks; and servers can enforce precise logic. The challenge is determining how to ***automatically decompose*** a single security function into a sequence of increasingly precise layers while respecting each device's constraints. One illustrative idea is a ***Sieve***-like pipeline for volumetric attack mitigation, where a switch implements an approximate first line of defense, a smartNIC further refines decisions, and the end host performs the final exact classification [11]. My goal is to generalize this into a ***compiler-driven framework*** that takes as input a high-level security specification and a topology of available devices, and automatically generates a layered defense tailored to each platform's capabilities. Such automation would bring principled, high-performance security to networks facing increasingly sophisticated adversaries.

AI for Network Control. A third direction of my future work is to bring ***reliable, interpretable, and generalizable AI*** into the network control loop. Despite the transformative impact of AI in many domains, networking has been slow to adopt AI-driven control because of three persistent challenges: models often fail to generalize across the enormous diversity of network conditions, their decisions can be difficult to interpret for human operators, and the cost of mistakes in a live network is high. My research trajectory positions me to address all three barriers. First, the programmable data plane provides an unprecedented opportunity to generate rich network ***features*** at line rate, enabling AI models to be trained and evaluated on traffic that spans application domains, protocol behavior, and network conditions. Second, safety can be ensured by monitoring the network's real-time performance metrics and validating whether the AI's actions are achieving the intended outcome (e.g., reducing delay or congestion). When deviations exceed predefined safety thresholds learned during demystification studies, the system can immediately revert to conservative, rule-based control strategies implemented on programmable hardware. Third, interpretability can be improved by running AI-driven inference across large-scale simulations on a heterogeneous testbed that incorporate diverse applications, network types (cellular, Wi-Fi, wired), and programmable devices, allowing operators to understand, audit, and refine AI behavior before deployment. By unifying these ingredients, my goal is to make AI not just an add-on for network automation, but a ***trustworthy and integral*** component of closed-loop network control.

5.2 Impact, Engagement, and Societal Benefit

I plan to broaden the real-world reach and societal value of my research by working directly with service providers, cloud platforms, and network operators to translate new research ideas into practical systems. My existing work on demystifying applications, continuous measurement, attack detection, and scaling up Internet infrastructure provides early evidence that I care about building technologies that address concrete operational needs. Moving forward, I aim to continue designing systems that can be incorporated into cloud deployments and enterprise/campus networks. I also plan to collaborate with network operators to evaluate and refine these systems in production environments. Further, I plan to pursue commercialization opportunities for components

that improve performance and security or reduce operational cost. Alongside these efforts, I intend to contribute to networking more broadly by working with policy communities that focus on transparency, accountability, and the security of critical Internet services. I will continue to release open-source software, datasets, and experimental testbeds that make advanced networking tools widely accessible. Through these combined efforts, I hope to ensure that improvements in network observability and control lead to long-lasting positive impact.

References

[1] Satadal Sengupta, Hyojoon Kim, and Jennifer Rexford. Continuous in-network round-trip time monitoring. In ***Proceedings of the ACM SIGCOMM 2022 Conference***, pages 473–485, 2022.

[2] Oliver Michel, Satadal Sengupta, Hyojoon Kim, Ravi Netravali, and Jennifer Rexford. Enabling passive measurement of Zoom performance in production networks. In ***Proceedings of the 22nd ACM Internet Measurement Conference (IMC)***, pages 244–260, 2022.

[3] Satadal Sengupta, Hyojoon Kim, Daniel Jubas, Maria Apostolaki, and Jennifer Rexford. Passive data-plane telemetry to mitigate long-distance BGP hijacks. ***New Ideas in Networked Systems (NINeS)***, 2026.

[4] Oliver Michel, Satadal Sengupta, Hyojoon Kim, Ravi Netravali, and Jennifer Rexford. Scalable video conferencing using SDN principles. In ***Proceedings of the ACM SIGCOMM 2025 Conference***, pages 1213–1231, 2025.

[5] Satadal Sengupta, Niloy Ganguly, Pradipta De, and Sandip Chakraborty. Exploiting diversity in Android TLS implementations for mobile app traffic classification. In ***The World Wide Web Conference (The WebConf, formerly WWW)***, pages 1657–1668, 2019.

[6] Abhijit Mondal, Satadal Sengupta, Bachu Rikith Reddy, MJV Koundinya, Chander Govindarajan, Pradipta De, Niloy Ganguly, and Sandip Chakraborty. Candid with Youtube: Adaptive streaming behavior and implications on data consumption. In ***Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)***, pages 19–24, 2017.

[7] Rijurekha Sen, Hasnain Ali Pirzada, Amreesh Phokeer, Zaid Ahmed Farooq, Satadal Sengupta, David Choffnes, and Krishna P Gummadi. On the free bridge across the digital divide: Assessing the quality of Facebook's Free Basics service. In ***Proceedings of the 2016 Internet Measurement Conference (IMC)***, pages 127–133, 2016.

[8] Satadal Sengupta, Niloy Ganguly, Sandip Chakraborty, and Pradipta De. HotDASH: Hotspot aware adaptive video streaming using deep reinforcement learning. In ***2018 IEEE 26th International Conference on Network Protocols (ICNP)***, pages 165–175. IEEE, 2018.

[9] Satadal Sengupta, Vinay Kumar Yadav, Yash Saraf, Harshit Gupta, Niloy Ganguly, Sandip Chakraborty, and Pradipta De. MoViDiff: Enabling service differentiation for mobile video apps. In ***2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)***, pages 537–543. IEEE, 2017.

[10] Siddharth Singh, Vedant Nanda, Rijurekha Sen, Satadal Sengupta, Ponnurangam Kumaraguru, and Krishna P Gummadi. Leveraging facebook's free basics engine for web service deployment in developing regions. In ***Proceedings of the Ninth International Conference on Information and Communication Technologies and Development (ICTD)***, pages 1–11, 2017.

[11] Sophia Yoo, Satadal Sengupta, Maria Apostolaki, and Jennifer Rexford. Sieve: Layered network defenses against large-scale attacks. ***P4 Workshop***, 2023.