Capture and Analysis of Traffic Traces on a Wide-Area NDN Testbed

Sankalpa Timilsina
Tennessee Technological University
Cookeville, TN, USA
stimilsin43@tntech.edu

Davide Pesavento
National Institute of Standards and Technology
Gaithersburg, MD, USA
davide.pesavento@nist.gov

Junxiao Shi
National Institute of Standards and Technology
Gaithersburg, MD, USA
junxiao.shi@nist.gov

Susmit Shannigrahi
Tennessee Technological University
Cookeville, TN, USA
sshannigrahi@tntech.edu

Lotfi Benmohamed
National Institute of Standards and Technology
Gaithersburg, MD, USA
lotfi.benmohamed@nist.gov

ABSTRACT

High-quality network traffic measurements from realistic network deployments are crucial to analyze and better understand emerging network technologies for the purpose of maturing them. However, achieving this measurement goal for the Named Data Networking (NDN) protocol remains a challenge mainly due to the lack of real-world deployments. To address this gap, we have created a dataset of NDN traffic traces and a software toolkit for capturing, analyzing, and replaying these traces. Our dataset, obtained directly from the real routers of the official NDN testbed, is the first non-synthetic dataset of this scale openly available to the research community. This paper presents the dataset and the tools, discusses its properties, and shares insights applicable to other NDN research.

CCS CONCEPTS

• **Networks** → **Network experimentation**; **Network measurement**; *Network layer protocols*; *Wide area networks.*

KEYWORDS

Named data networking, packet trace, traffic analysis, testbed, wide area network, realistic traffic, file transfer, video streaming

ACM Reference Format:
Sankalpa Timilsina, Davide Pesavento, Junxiao Shi, Susmit Shannigrahi, and Lotfi Benmohamed. 2023. Capture and Analysis of Traffic Traces on a Wide-Area NDN Testbed. In *10th ACM Conference on Information-Centric Networking (ACM ICN '23), October 9–10, 2023, Reykjavik, Iceland.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3623565.3623707

1 INTRODUCTION

Named Data Networking (NDN) [46] is a next generation Internet architecture that has garnered significant attention and exploration over the past decade. Despite the relatively long past, there is a distinct lack of open datasets that can be used for NDN research and experimentation. The lack of open datasets containing realistic NDN traffic traces poses a considerable challenge for researchers and practitioners looking to experiment with and evaluate NDN's capabilities, thus hindering its adoption. Without realistic packet traces, one cannot strongly demonstrate the benefits of NDN deployment for real-world use cases.

Our work aims to bridge this crucial gap by providing a comprehensive and publicly available NDN traffic dataset. The initial version of this growing dataset consists of 21 hours of NDN packet traces spanning 7 different scenarios, and encompassing an extensive volume of 320 GiB of traffic representing more than 108 million packets. This dataset is collected on the official NDN testbed [27]. It is worth highlighting that the NDN testbed stands as the most proximate representation of an authentic, real-world NDN ecosystem presently accessible to us. While its scale is limited, the testbed carries real application and background traffic and has rich connectivity across geographical regions. By offering researchers access to such data-rich repository, we hope to empower them to explore and investigate NDN's true potential, enabling more accurate evaluations with reproducible results and fostering innovation in the NDN ecosystem. Our preliminary analysis of the traces demonstrates the effectiveness of NDN's in-network caching and provides insights into the statistical distribution of name length, `InterestLifetime`, `FreshnessPeriod`, and `HopLimit` values.

Along with the raw dataset, we also provide a comprehensive set of tools and scripts to support the utilization of the dataset in NDN research (see Appendix A for the full list). These tools include an efficient NDN traffic dumping tool with online compression and anonymization features, ensuring privacy and scalability of the collected data. Additionally, we offer a collection of programs for analyzing NDN traces, extracting packet-level metrics, and visualizing the findings. Lastly, we present a pair of proof-of-concept consumer/producer applications designed for the ndnSIM simulator [24], enabling controlled experimentation with the traces. Researchers can leverage these applications to replay NDN packet traces, facilitating in-depth investigation and evaluation of the protocol's behavior and performance.

Through these contributions, we aspire to not only fill the void of NDN traffic datasets but ultimately aim to accelerate the adoption

ACM ICN '23, October 9–10, 2023, Reykjavik, Iceland

ACM ISBN 979-8-4007-0403-1/23/10…$15.00
https://doi.org/10.1145/3623565.3623707

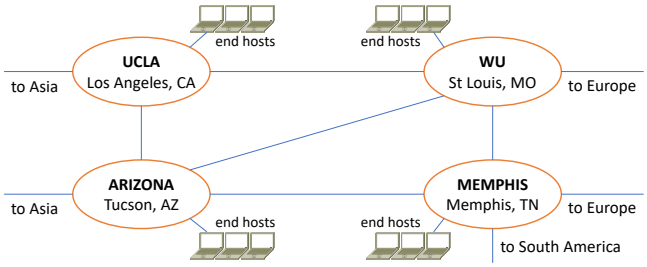

Figure 1: NDN testbed in North America.

and realization of NDN's potential, paving the way for a more efficient, secure, and content-centric Internet.

2 MOTIVATION AND RELATED WORK

The potential benefits of NDN have attracted significant interest from researchers and practitioners in the networking field. However, the scarcity of NDN deployments in production networks limits the availability of real-world data for performance evaluations. Synthetic traffic or traffic patterns derived from IP traces provide very limited insights, hampering the development of efficient networking solutions.

There are numerous works that investigated NDN for different networking use cases. These include the Internet of Things [2, 21, 31, 34, 48], video streaming [8, 9, 37, 38], content delivery networks [12, 13, 43], real-time video [14, 15, 22], scientific datasets [5, 11, 35, 45], network monitoring [17, 29, 30], and other applications [20, 23, 32, 36]. These solutions adopt various methods to acquire and publish packet traces for NDN research. Some rely on IP traffic from existing networks, capturing and potentially converting it to NDN traffic for analysis. Although these solutions provide readily available packet traces, they are not NDN-based and require additional effort to interpret the results. On the other hand, there are approaches [8, 9, 13, 35] that use native NDN networks to generate the data traces. While these solutions contribute valuable insights to NDN research, they do not provide a comprehensive set of standardized NDN traces for broader experimentation and evaluation.

To address this need, we have developed a comprehensive toolkit to generate and capture NDN traffic. Leveraging the global NDN testbed [27] for packet transport and NSF's FABRIC [6] testbed to deploy NDN applications, we have collected a valuable dataset of NDN traces. In this paper, we present our toolkit and describe the dataset's composition and potential applications for research. By sharing this dataset with the community, we hope to promote further analysis and performance studies of NDN networks.

3 METHODOLOGY

The NDN project operates a worldwide research testbed [27] which, as of June 2023, has 16 routers and 27 links across 10 countries and 4 continents. It operates as an overlay network interconnected via UDP tunnels over the IPv4 Internet. Each router runs the *NDN Forwarding Daemon* (NFD) [4], the *NDN Link State Routing* (NLSR) daemon [19], and the *nginx* server for accepting WebSocket connections over HTTPS. Anyone can connect to the NDN testbed via UDP or WebSockets.

We installed traffic dumpers (section 3.2) on four testbed routers in North America (fig. 1) and collected traffic traces over a duration of 3 hours every day for 7 days in May-June 2023. At the same time, we ran two real NDN applications (section 3.1) developed by the community, to supplement existing traffic from NLSR and other testbed users.

3.1 NDN Applications for Realistic Traffic Generation

Although the global NDN testbed is a public network that may be used by anyone, its adoption is limited and the traffic volume is usually low. To increase the volume of traffic, we ran two NDN applications, file transfer and video streaming, on virtual machines provided by NSF's FABRIC testbed [6]. Our traffic traces contain the traffic generated by these applications, as well as existing traffic from NLSR and other testbed users.

3.1.1 File Transfer Application. The file transfer application employs NDN-DPDK [40] as the producer and the NDNc client [45] as the consumer. The producer node runs an NDN-DPDK forwarder and an NDN-DPDK file server. The forwarder opens a UDP socket toward a testbed router and periodically sends prefix registration commands for the prefix `/fileserver.{random-number}`. The file server process connects to the local forwarder via *memif* transport.

Eight files, with sizes ranging from 1 MiB to 2 GiB, are placed on an NVMe drive on the file server. Each file is named after the producer's prefix followed by the file size, e.g., `/fileserver.1685589141/500M.bin`. Upon receiving an Interest for a file segment, the file server replies with a Data packet containing the requested portion of the file, which is up to 6 KiB in length. The Data packet can then be cached at the local NDN-DPDK forwarder as well as other NDN routers in the network.

The consumer node runs an NDN-DPDK forwarder and an NDNc file transfer client. The forwarder opens a UDP socket toward an NDN testbed router and sets it as the default route. The NDNc file transfer client connects to the local forwarder via *memif* transport and retrieves files of random sizes at random intervals. File retrievals use the AIMD congestion control algorithm.

3.1.2 Video Streaming Application. The video streaming application employs NDN-DPDK as the producer and the NDNts adaptive video player [38, 39] as the consumer. The producer setup is the same as the file transfer application, except that the producer prefix is changed to `/videoserver.{random-number}` and the NVMe drive contains video files pre-encoded in DASH format [1]. We prepared seven videos with durations between 1 minute and 30 minutes. Each video is named after the producer's prefix followed by the video duration, e.g., `/videoserver.1685589712/10-min.mp4`.

The consumer node runs the NDNts video player in a headless Chromium browser controlled by Puppeteer [42]. Our script launches the browser, opens the web-based video player, connects to a testbed router via WebSocket, plays a specified video, and finally closes the browser upon playback completion or timeout.

Each DASH video is encoded into hundreds of "video segment" files. While playing a DASH video involves retrieving these files, it differs from file transfer in that: (1) the video player can switch to a higher or lower video resolution based on measured available

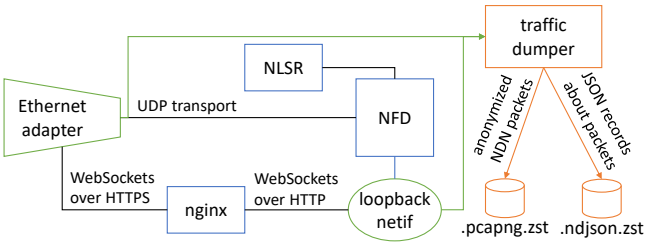

Figure 2: Router software with traffic dumper.

bandwidth; (2) each individual file retrieval adopts a CUBIC-like congestion control algorithm, instead of the AIMD algorithm.

3.1.3 NLSR Routing. NLSR [19], operating in hyperbolic routing mode [18], provides routing across the testbed network. It operates as an NDN application that runs on every testbed router, exchanging routing information with adjacent routers via Interest and Data packets, and controlling the local NFD forwarder via management commands.

NLSR instances exchange `HELLO` packets between adjacent nodes to detect link state. They collectively run the PSync [47] data synchronization protocol to establish a common view of an "LSDB" dataset that describes the name and hyperbolic coordinates of each router as well as what prefixes are advertised, which allows each router to compute the routing table accordingly. In PSync, each node periodically sends a multicast Interest that contains a digest of the current dataset contents. In the steady state, this Interest would not be answered and would eventually time out. This Interest is replied to only if an NLSR instance wants to update the dataset (e.g., advertise a new prefix), in which case the reply will contain the name of a separate Data packet that carries the full update. After that, other NLSR instances will send regular Interests to retrieve the full update and may send additional Interests to fetch the certificates required to validate Data signatures.

Our traces include the routing traffic between NLSR daemons on different hosts, but exclude the management commands between NLSR and the local NFD. While these commands happen to use Interest-Data exchanges, they are not transmitted over a network link, but are instead used as a form of secure inter-process communication. This kind of management plane traffic is comparable to the netlink messages sent to/from the Linux kernel, and we therefore consider it out of scope for this work.

3.2 Traffic Capturing

We developed a traffic dumper, *ndntdump*, for capturing NDN packets traversing an NDN testbed router. *ndntdump* can either capture Ethernet traffic from network interface cards via an `AF_PACKET` socket, or read packet traces from PCAP files created by other tools such as *tcpdump*. For each Ethernet frame, *ndntdump* performs up to three online processing steps (the second and third steps can optionally be disabled).

(1) **NDN packet extraction.** The Ethernet frame is decoded into protocol layers using the powerful GoPacket [7] library. If a valid NDN or NDNLPv2 packet [26, 28] is found in the UDP, TCP, or WebSocket payload, the frame is accepted. Otherwise, the frame is discarded.

(2) **Address anonymization.** To improve privacy, *ndntdump* performs XOR-based anonymization on MAC address and IP address fields in Ethernet/IPv4/IPv6 headers as well as the `X-Forwarded-For` HTTP header on the WebSocket `Upgrade` request. For each IPv4/IPv6 address, the leading 24/48 bits are kept, so that an analyst may cross-reference the IP subnet with BGP and geolocation data but cannot identify individual hosts. As part of this work, we offer a basic level of anonymization, which might not be appropriate in all situations. If necessary, users have the option to augment our toolkit with any additional state-of-the-art anonymization technique [10].

(3) **Payload blanking.** Payload fields, such as the `ApplicationParameters` field in Interest packets and the `Content` field in Data packets, account for most of the packet length but their values are rarely of interest in traffic analysis. Therefore, *ndntdump* replaces the TLV value of each payload field with a string of zeros. This operation does not change the TLV length of the zeroed-out fields or the overall packet length, but drastically improves the compressibility of the output files. On the other hand, this operation would change the implicit digest of a Data packet, so that an analyst who wants to match Data packets against full-name Interests must perform the match with exact-name only and always assume the implicit digest component matches the Data packet.

After online processing, *ndntdump* produces two output files: a "`packets`" file and a "`records`" file. The `packets` file contains Ethernet frames that carry NDN traffic. It is written in pcapng format [44] and compressed with the zstd algorithm. Each packet contains the same protocol headers as captured, after applying address anonymization and payload blanking. The `records` file contains a machine-readable description of each packet. It is written in newline-delimited JSON (NDJSON) format [16] and compressed with zstd. Each packet is represented by a JSON object that contains the fields described in Table 1.

Table 1: Packet information available in JSON format.

Packet direction	Incoming or Outgoing
Flow key	String identifying UDP/TCP flow
Packet type	Interest, Data, Nack, or Fragment
Packet size	At NDNLPv2 layer and at network layer
Interest packet fields	Name, CanBePrefix, MustBeFresh, ForwardingHint, InterestLifetime, HopLimit
Data packet fields	Name, ContentType, FreshnessPeriod, FinalBlockId
Nack packet fields	NackReason, plus the fields from the enclosed Interest

3.2.1 Handling Fragments. NDN nodes perform hop-by-hop fragmentation and reassembly [3] so that packets larger than the link MTU can be transported. Packets larger than the MTU are fragmented by NFD per the NDNLPv2 protocol [28, 41] before transmission on the wire, which is then captured by *ndntdump*. To recover the whole Interest/Data packet, it is necessary to reassemble those fragments. However, performing reassembly in *ndntdump* would significantly increase RAM usage and exceed the RAM budget we

are allowed to use on the testbed routers. Therefore, we took a different approach to extract the packet fields. For an NDNLPv2 packet that carries the first fragment of an Interest/Data packet, *ndntdump* parses the packet with a truncation-tolerant TLV decoder. According to the NDN packet format [26], the most interesting fields (i.e., those in the `records` output file) of an Interest/Data packet appear at the front of the packet before payload and signature fields. As long as the Name is not too long, we can extract the desired information using this technique. A trade-off is that *ndntdump* cannot identify the payload fields in subsequent fragments, so that it cannot perform payload blanking and the output file becomes less compressible.

3.2.2 Deployment. We deployed *ndntdump* as a Docker container on four NDN testbed routers in North America (fig. 1). In our deployment, the capture source includes not only the Ethernet adapter through which NFD communicates with other routers and end hosts over UDP, but also the loopback network interface that carries decrypted WebSocket traffic between nginx and NFD (fig. 2). The traffic dumpers were scheduled to run between 05:00 and 08:00 UTC every day.

4 DATASET DESCRIPTION AND USAGE

The dataset was collected over a duration of 3 hours every day for 7 days. On each day, we arranged the file transfer and video streaming applications to create different traffic scenarios:

A. Baseline traffic from NLSR and other testbed users.
B. File transfer, with producer connected to UCLA and consumer connected to WU.
C. Video streaming, with producer connected to UCLA and consumer connected to WU.
D. Both applications, with producers connected to UCLA and consumers connected to WU.
E. Both applications, with producers connected to UCLA and consumers connected to WU, MEMPHIS, ARIZONA.
F. Both applications, with producers connected to UCLA, MEMPHIS, ARIZONA and consumers connected to WU.
G. Both applications, with producers connected to UCLA and consumers connected to WU. In this scenario, end hosts are located at multiple FABRIC sites so that they have different latencies to the testbed routers.

Each day's trace is stored in four zstd-compressed pcapng files, containing the packets captured on the four chosen testbed routers. Table 2 lists the basic statistics of the traces, grouped by scenario. The packet counts reflect the totals from all four routers together. "Size" is the total traffic volume at the NDNLPv2 layer. "CBP" and "MBF" denote the percentage of Interests carrying the `CanBePrefix` and `MustBeFresh` flags, respectively.

4.1 Traffic Analysis

This section presents a preliminary analysis of the collected packet traces.

4.1.1 Name Prefixes. Figure 3 shows the name length in terms of components and bytes. It is derived from the packet trace of scenario E at the ARIZONA router, where a subset of consumers for both file transfer and video streaming applications are connected.

Table 2: Basic statistics of the traces for each scenario.

	Total packet count			Size	CBP	MBF
	Interests	Data	Nacks	(MiB)	(%)	(%)
A	149,683	84,617	10,699	62	59.55	81.26
B	10,041,803	9,906,857	35,350	60,604	1.23	1.56
C	1,533,686	1,074,899	11,388	3,987	24.02	25.95
D	10,352,712	10,152,325	10,360	60,858	2.24	2.74
E	6,353,608	5,908,883	15,645	34,487	3.87	4.81
F	14,823,353	14,535,375	6,899	87,706	1.59	1.94
G	10,330,511	13,271,744	7,015	80,548	1.59	2.18

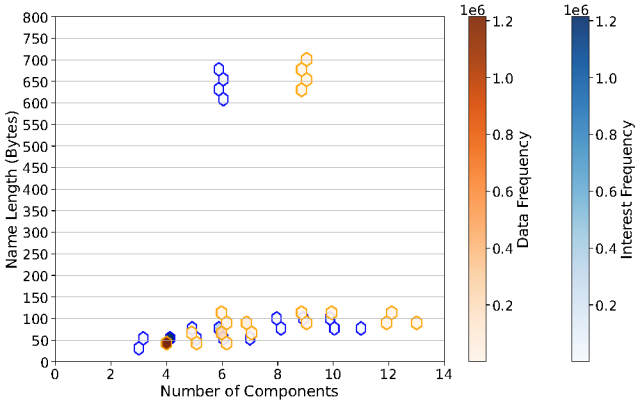

Figure 3: Name length distribution at ARIZONA router.

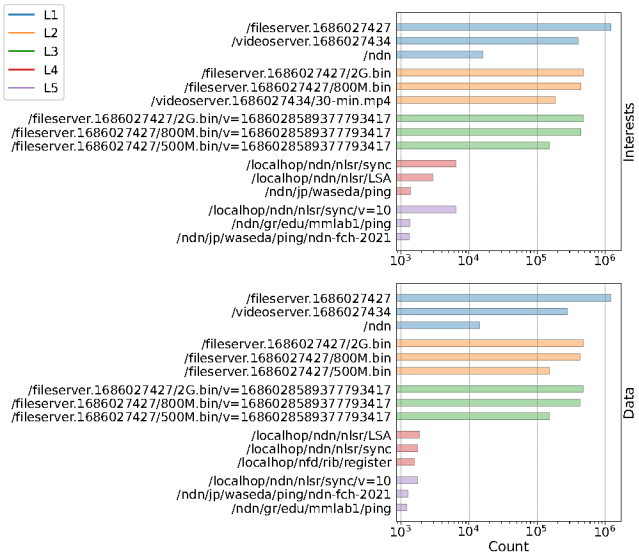

Figure 4: Popular name prefixes at ARIZONA router.

We can see that most names have either 4 or 6 components, which can be attributed to the names used by our two applications. Other names are scattered between 3 and 14 components in length, which

may include those from NLSR and other testbed users. In terms of bytes, most names fall between 20 and 130 bytes. A small number of names are over 600 bytes long, which can be attributed to the PSync Interests and replies sent by NLSR.

Figure 4 shows the three most popular name prefixes, derived from the same packet trace. We see that our two applications have taken the top 2 or 3 spots when aggregated by the leading 1, 2, or 3 components of their name prefixes. Looking at longer prefixes, we have observed various names used by NFD prefix registration commands, routing protocol traffic, and *ndnping* traffic for testbed monitoring purposes.

4.1.2 Throughput. Figure 5 shows the network throughput over time in terms of packets and Mbps in scenario E. This scenario has both file transfer and video streaming applications, all producers are connected to the UCLA router, and consumers are spread among WU, MEMPHIS, and ARIZONA. We can see that the traffic at UCLA is lower than the sum of the traffic measured on its adjacent routers WU and ARIZONA (MEMPHIS is not a neighbor, see fig. 1). This suggests that NDN in-network caching has been effective in satisfying some Interests locally without forwarding them to the producer.

4.1.3 CanBePrefix and MustBeFresh. The columns labeled "CBP" and "MBF" in Table 2 show the percentage of Interests that carry the `CanBePrefix` and `MustBeFresh` flags, respectively. This kind of Interests are often used for name discovery. We can see that these percentages are high in the baseline scenario A, but are substantially lower in all scenarios that include file transfers, because the file transfer consumer uses these flags sparingly.

4.1.4 InterestLifetime and FreshnessPeriod. Figure 6 shows the distribution of `InterestLifetime` and non-zero Data `FreshnessPeriod`, aggregated over all 7 days of traces. We see that 2000 ms is the most popular `InterestLifetime`. Values just under 2000 ms are also frequently observed; this is likely caused by NDN-DPDK forwarders decrementing the `InterestLifetime` field by the amount of time the packet spent in the queues. The default `InterestLifetime` of 4 seconds is also common.

The most frequent `FreshnessPeriod` value by far is zero, which accounts for 97.3 % of Data packets but has been excluded from the plot. Other popular values are 1 ms, 1 second, 10 seconds, and 30 minutes.

4.1.5 HopLimit. Figure 7 shows the distribution of Interest `HopLimit`, aggregated over all traces. We see three peaks at and below 32, 64, and 255. These can be attributed to the default `HopLimit` settings in various NDN implementations.

4.1.6 NLSR Data Packet Sizes. Figure 8 shows the distribution of Data packet sizes for NLSR traffic (Data name contains the component "`nlsr`"), aggregated over all 7 days of traces. We observe four peaks around 300, 900, 1700, and 3500 bytes. These can respectively be attributed to: (1) `HELLO` replies and single LSDB records, (2) PSync replies, (3) full LSDB coordinates dataset, (4) full LSDB names dataset.

4.2 Using the Dataset in Simulations

We expect that one of the main uses of the dataset will be to drive simulations aimed at studying the behavior of NDN networks and improving the state of the art in various areas, from forwarding strategies to caching policies, from cybersecurity to performance optimizations. To facilitate that, we developed a pair of proof-of-concept consumer/producer applications for ndnSIM [24], one of the most popular NDN simulators used by the research community. These simulated applications are driven by *ndntdump*'s `records` output file (see section 3.2). The consumer can replay Interests at the proper time offsets, with the same `InterestLifetime` and `MustBeFresh` values as in the traffic trace. The producer will respond to those Interests whose name appears in the trace, but only supports exact matching and does not take any other Data fields from the trace into account. Our plan for the future is to improve these programs to reproduce the traffic more faithfully.

5 DISCUSSION AND FUTURE WORK

Capturing a realistic traffic trace of an experimental protocol such as NDN presents some unique challenges. While our dataset strives to cover the most common use cases encountered in NDN-related academic literature, it also comes with a few limitations that the user should be aware of.

Diversity of applications. There are only a handful of NDN applications today that are sufficiently mature and that could be used to emulate a realistic usage of the network. Indeed, while a fair amount of NDN-based software has been developed over the years, a significant portion of it has a narrow scope, primarily driven by the authors' research needs, and consequently tends to be abandoned soon after reaching its goals. Once that happens, the software rapidly succumbs to bit rot and becomes unusable. It is our hope that, as the NDN community grows and the protocol stabilizes, maintaining the software over longer periods of time will become less burdensome and a wider array of applications will be readily available to create more diversified traffic traces. We are currently evaluating additional types of applications for inclusion in future trace collections.

Traffic mix. While we used real NDN applications in all our collection scenarios, we do not claim that the resulting traffic mix is representative of an hypothetical future NDN network. It is well known [33] that today's Internet traffic is largely dominated by video streaming, with web browsing and downloads (games, mobile app stores, file sharing) a distant second. However, this does not automatically imply that an "NDN-based Internet" will exhibit similar patterns of network usage. In fact, certain key features of NDN, such as caching and multicasting, have the potential to drastically reduce the volume of traffic generated by applications that are traditionally considered bandwidth hungry [8, 9]. The proliferation of applications based on Sync protocols [25, 47] can further alter the traffic composition in novel ways. As we gain more experience from deployments of NDN in the wild, we plan to adjust our methodology to more closely mimic the natural traffic mix seen in real environments.

Traffic anonymization. While the current anonymization approach we have employed is relatively basic and "best effort", this was a

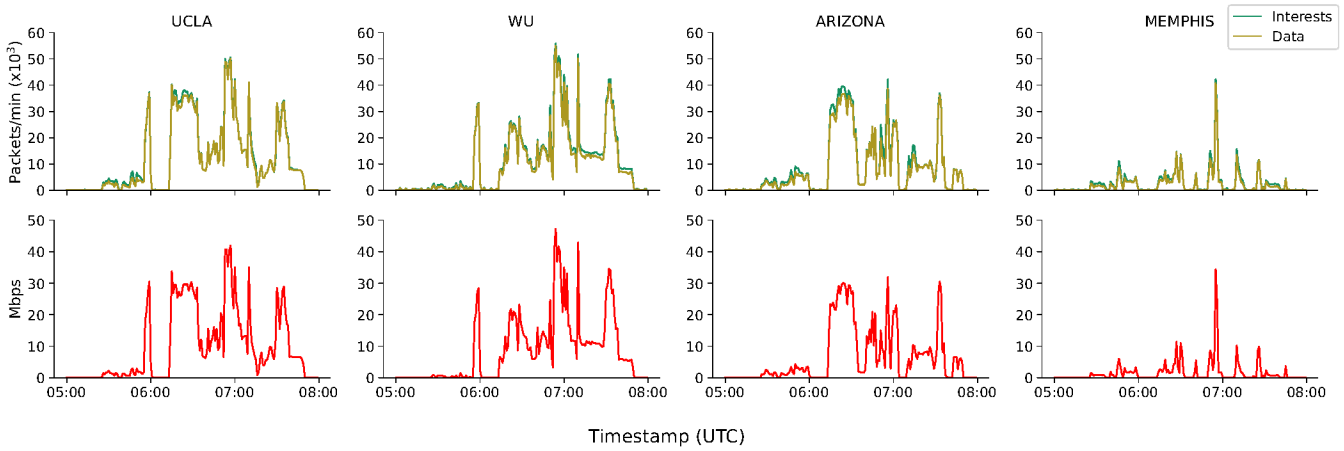

Figure 5: Throughput measured in scenario E at the 4 routers.

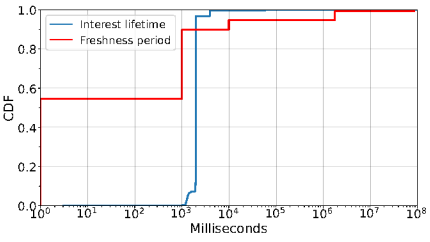

Figure 6: Cumulative distribution of InterestLifetime and FreshnessPeriod values across all traces.

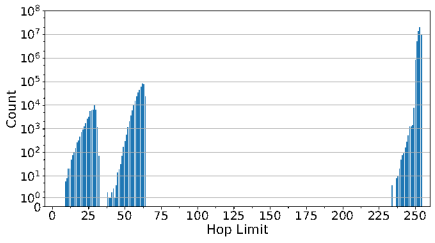

Figure 7: Distribution of Interest HopLimit values across all traces.

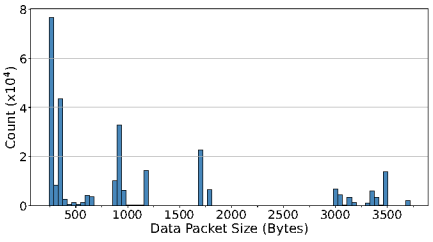

Figure 8: Distribution of NLSR Data packet sizes across all traces.

conscious choice. This decision sought to keep the complexity of the dumper's implementation in check, especially considering that the primary focus of our paper does not lie in the intricacies of anonymization techniques. We note that the task of anonymizing NDN traffic could very well be a rich topic for a separate in-depth study. Moving forward, we intend to explore this area as a prospective avenue for further NDN research.

Testbed scale. As explained in section 3, the NDN testbed operates as an overlay network over the public Internet, with routers in four continents. On the other hand, FABRIC is primarily U.S.-centric and currently provides only one Internet peering point, located in Washington, DC. Therefore, we limited our traffic collection points to the routers located in North America, to avoid inefficient routing over intercontinental links that would be atypical in a real network. We are actively exploring the use of additional testbeds and commercial cloud providers as a platform to run larger scale collection campaigns on multiple continents, with consumers and producers spread across the globe.

6 CONCLUSION

This paper describes our attempt at creating a framework and associated tools to generate realistic NDN traffic traces and demonstrate applicability to NDN research. The developed tools include (a) an efficient NDN traffic capture tool with compression and anonymization features, as well as associated scripts that automate the collection of traffic traces, and (b) a set of programs that facilitate the analysis of NDN traces and allow for the extraction and visualization of common traffic metrics. Leveraging the NDN testbed and NSF's FABRIC, we deployed our tools and collected real-world NDN traffic traces. An initial analysis of the traces shows the potential for providing valuable insights to NDN researchers.

This initial effort helped us identify possible next steps, such as including additional types of applications in future trace collections and carrying out longer captures on a wider scale, with application endpoints spread across the globe. Our roadmap also includes more streamlined sharing of datasets with the community, in particular a portal that will allow others to contribute their own traces and a web-based frontend to easily visualize the data. We believe that this collaborative capability will enhance, among other things, the reproducibility of NDN research results.

DISCLAIMER

Any mention of commercial products or reference to commercial organizations is for information only; it does not imply recommendation or endorsement by NIST, nor does it imply that the products mentioned are necessarily the best available for the purpose.

ACKNOWLEDGMENTS

We would like to thank Alex Afanasyev for his help in deploying the traffic capture software to the NDN testbed. We also thank the anonymous reviewers and our shepherd, Eric Osterweil, for their comments and suggestions. This work has been supported by NSF Awards 2126148, 2019012, and 2019163.

REFERENCES

[1] ISO/IEC JTC 1/SC 29. 2019. *Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats.* Technical Report. ISO/IEC 23009-1:2019. https://www.iso.org/standard/79329.html

[2] Amar Abane, Mehammed Daoui, Samia Bouzefrane, Soumya Banerjee, and Paul Mühlethaler. 2020. A realistic deployment of named data networking in the internet of things. *Journal of Cyber Security and Mobility* 9, 1 (2020).

[3] Alexander Afanasyev, Junxiao Shi, Lan Wang, Beichuan Zhang, and Lixia Zhang. 2015. *Packet Fragmentation in NDN: Why NDN Uses Hop-By-Hop Fragmentation (NDN Memo).* Technical Report NDN-0032. NDN.

[4] Alexander Afanasyev, Junxiao Shi, Beichuan Zhang, Lixia Zhang, Ilya Moiseenko, Yingdi Yu, Wentao Shang, Yanbiao Li, Spyridon Mastorakis, Yi Huang, Jerald Paul Abraham, Eric Newberry, Steve DiBenedetto, Chengyu Fan, Christos Papadopoulos, Davide Pesavento, Giulio Grassi, Giovanni Pau, Hang Zhang, Tian Song, Haowei Yuan, Hila Ben Abraham, Patrick Crowley, Syed Obaid Amin, Vince Lehman, Muktadir Chowdhury, and Lan Wang. 2021. *NFD Developer's Guide.* Technical Report. NDN-0021, Revision 11. https://named-data.net/publications/techreports/ndn-0021-11-nfd-guide/

[5] Mahdieh Ahmadi, James Roberts, Emilio Leonardi, and Ali Movaghar. 2019. Poster: Impact of traffic characteristics on request aggregation in an NDN router. In *2019 IFIP Networking Conference (IFIP Networking).* IEEE, 1–2. https://doi.org/10.23919/IFIPNETWORKING46909.2019.8999462

[6] Ilya Baldin, Anita Nikolich, James Griffioen, Indermohan Inder S Monga, Kuang-Ching Wang, Tom Lehman, and Paul Ruth. 2019. FABRIC: A national-scale programmable experimental network infrastructure. *IEEE Internet Computing* 23, 6 (2019), 38–47.

[7] Graeme Connell, Nigel Tao, Cole Mickens, Ben Daglish, Luis Martinez, Remco Verhoef, Hiroaki Kawai, Lukas Lueg, Laurent Hausermann, Bill Green, Christian Mäder, Gernot Vormayr, Vitor Garcia Graveto, Elias Chavarria Reyes, and Daniel Rittweiler. 2022. *google/gopacket: Provides packet processing capabilities for Go.* Retrieved June 12, 2023 from https://github.com/google/gopacket

[8] Ishita Dasgupta, Susmit Shannigrahi, and Michael Zink. 2021. A hybrid NDN-IP Architecture for Live Video Streaming: A QoE Analysis. In *2021 IEEE International Symposium on Multimedia (ISM).* 148–157. https://doi.org/10.1109/ISM52913.2021.00032

[9] Ishita Dasgupta, Susmit Shannigrahi, and Michael Zink. 2022. A Hybrid NDN-IP Architecture for Live Video Streaming: From Host-Based to Content-Based Delivery to Improve QoE. *International Journal of Semantic Computing* 16, 02 (2022), 163–187.

[10] Niels Van Dijkhuizen and Jeroen Van Der Ham. 2018. A Survey of Network Traffic Anonymisation Techniques and Implementations. *ACM Comput. Surv.* 51, 3, Article 52 (may 2018), 27 pages. https://doi.org/10.1145/3182660

[11] Chengyu Fan, Susmit Shannigrahi, S. DiBenedetto, C. Olschanowsky, C. Papadopoulos, and H. Newman. 2015. Managing scientific data with named data networking. *NDM '15* (2015). https://doi.org/10.1145/2832099.2832100

[12] Chavoosh Ghasemi, Hamed Yousefi, and Beichuan Zhang. 2020. Far Cry: Will CDNs Hear NDN's Call?. In *Proceedings of the 7th ACM Conference on Information-Centric Networking (ICN '20).* Association for Computing Machinery, 89–98. https://doi.org/10.1145/3405656.3418708

[13] Chavoosh Ghasemi, Hamed Yousefi, and Beichuan Zhang. 2020. ICDN: An NDN-Based CDN. In *Proceedings of the 7th ACM Conference on Information-Centric Networking (ICN '20).* Association for Computing Machinery, 99–105. https://doi.org/10.1145/3405656.3418716

[14] Peter Gusev and Jeff Burke. 2015. NDN-RTC: Real-Time Videoconferencing over Named Data Networking. In *Proceedings of the 2nd ACM Conference on Information-Centric Networking (ACM-ICN '15).* Association for Computing Machinery, 117–126. https://doi.org/10.1145/2810156.2810176

[15] P. Gusev, Zhehao Wang, J. Burke, Lixia Zhang, T. Yoneda, Ryota Ohnishi, and E. Muramoto. 2016. Real-Time Streaming Data Delivery over Named Data Networking. *IEICE Trans. Commun.* (2016). https://doi.org/10.1587/TRANSCOM.2015AMI0002

[16] Thorsten Hoeger, Finn Pauls, Paul Fitzpatrick, Mathieu Aubin, Jesse Jackson, and Chris Dew. 2022. *NDJSON - Newline delimited JSON.* Retrieved June 17, 2023 from https://github.com/ndjson/ndjson-spec

[17] Siham Khoussi, Davide Pesavento, Lotfi Benmohamed, and Abdella Battou. 2017. NDN-trace: a path tracing utility for named data networking. In *Proceedings of the 4th ACM Conference on Information-Centric Networking.* 116–122.

[18] Vince Lehman, Ashlesh Gawande, Beichuan Zhang, Lixia Zhang, Rodrigo Aldecoa, Dmitri Krioukov, and Lan Wang. 2016. An experimental investigation of hyperbolic routing with a smart forwarding plane in NDN. In *2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS).* 1–10. https://doi.org/10.1109/IWQoS.2016.7590394

[19] Vince Lehman, A K M Mahmudul Hoque, Yingdi Yu, Lan Wang, Beichuan Zhang, and Lixia Zhang. 2016. *A Secure Link State Routing Protocol for NDN.* Technical Report. NDN-0037, Revision 1. https://named-data.net/publications/techreports/ndn-0037-1-nlsr/

[20] Teng Liang, Ju Pan, and Beichuan Zhang. 2018. Ndnizing existing applications: Research issues and experiences. In *Proceedings of the 5th ACM Conference on Information-Centric Networking.* 172–183.

[21] Teng Liang, Zhongda Xia, Guoming Tang, Yu Zhang, and Beichuan Zhang. 2021. NDN in large LEO satellite constellations: a case of consumer mobility support. *Information-Centric Networking* (2021). https://doi.org/10.1145/3460417.3482970

[22] Teng Liang, Yang Zhang, Beichuan Zhang, Weizhe Zhang, and Yu Zhang. 2022. Low Latency Internet Livestreaming in Named Data Networking. In *Proceedings of the 9th ACM Conference on Information-Centric Networking (ICN '22).* Association for Computing Machinery, 177–179. https://doi.org/10.1145/3517212.3559488

[23] Xinyu Ma and Lixia Zhang. 2021. GitSync: Distributed Version Control System on NDN. In *Proceedings of the 8th ACM Conference on Information-Centric Networking (ICN '21).* Association for Computing Machinery, 121–123. https://doi.org/10.1145/3460417.3483372

[24] Spyridon Mastorakis, Alexander Afanasyev, and Lixia Zhang. 2017. On the evolution of ndnSIM: An open-source simulator for NDN experimentation. *ACM SIGCOMM Computer Communication Review* 47, 3 (2017), 19–33.

[25] Philipp Moll, Varun Patil, Lixia Zhang, and Davide Pesavento. 2021. Resilient Brokerless Publish-Subscribe over NDN. In *MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM).* 438–444. https://doi.org/10.1109/MILCOM52596.2021.9652885

[26] NDN Project. 2023. *NDN Packet Format Specification, version 0.3.* Retrieved June 12, 2023 from https://docs.named-data.net/NDN-packet-spec/0.3/

[27] NDN Project. 2023. *NDN Testbed.* Retrieved June 12, 2023 from https://named-data.net/ndn-testbed/

[28] NDN Project. 2023. *NDNLPv2: NDN Link Protocol, version 2.* Retrieved June 12, 2023 from https://redmine.named-data.net/projects/nfd/wiki/NDNLPv2

[29] Kathleen Nichols. 2019. Lessons learned building a secure network measurement framework using basic NDN. In *Proceedings of the 6th ACM Conference on Information-Centric Networking.* 112–122.

[30] Davide Pesavento, Omar Ilias El Mimouni, Eric Newberry, Lotfi Benmohamed, and Abdella Battou. 2017. A Network Measurement Framework for Named Data Networks. In *Proceedings of the 4th ACM Conference on Information-Centric Networking (ICN '17).* Association for Computing Machinery, 200–201. https://doi.org/10.1145/3125719.3132113

[31] Davide Pesavento, Junxiao Shi, Kerry McKay, and Lotfi Benmohamed. 2022. PION: Password-based IoT Onboarding Over Named Data Networking. In *ICC 2022 - IEEE International Conference on Communications.* 1070–1075. https://doi.org/10.1109/ICC45855.2022.9839088

[32] Tamer Refaei, Jamie Ma, Sean Ha, and Sarah Liu. 2017. Integrating IP and NDN through an Extensible IP-NDN Gateway. In *Proceedings of the 4th ACM Conference on Information-Centric Networking (ICN '17).* Association for Computing Machinery, 224–225. https://doi.org/10.1145/3125719.3132112

[33] Sandvine. 2023. *The Global Internet Phenomena Report, January 2023.* Retrieved June 12, 2023 from https://www.sandvine.com/phenomena

[34] Wentao Shang, Qiuhan Ding, Alessandro Marianantoni, Jeff Burke, and Lixia Zhang. 2014. Securing building management systems using named data networking. *IEEE Network* 28, 3 (2014), 50–56. https://doi.org/10.1109/MNET.2014.6843232

[35] Susmit Shannigrahi, Chengyu Fan, and C. Papadopoulos. 2017. *Request aggregation, caching, and forwarding strategies for improving large climate data distribution with NDN: a case study.* https://doi.org/10.1145/3125719.3125722

[36] Susmit Shannigrahi, Chengyu Fan, and Greg White. 2018. Bridging the ICN Deployment Gap with IPoC: An IP-over-ICN Protocol for 5G Networks. In *Proceedings of the 2018 Workshop on Networking for Emerging Applications and Technologies (NEAT '18).* Association for Computing Machinery, 1–7. https://doi.org/10.1145/3229574.3229575

[37] Junxiao Shi. 2021. *NDN Video Streaming over QUIC.* Retrieved June 15, 2023 from https://yoursunny.com/t/2021/NDN-video-QUIC/

[38] Junxiao Shi. 2021. *The Reality of NDN Video Streaming.* Retrieved June 12, 2023 from https://yoursunny.com/t/2021/NDN-video-reality/

[39] Junxiao Shi. 2023. *NDNts Adaptive Video.* Retrieved June 12, 2023 from https://github.com/yoursunny/NDNts-video

[40] Junxiao Shi, Davide Pesavento, and Lotfi Benmohamed. 2020. NDN-DPDK: NDN Forwarding at 100 Gbps on Commodity Hardware. In *Proceedings of the 7th ACM Conference on Information-Centric Networking (ICN '20).* Association for Computing Machinery, 30–40. https://doi.org/10.1145/3405656.3418715

[41] Junxiao Shi and Beichuan Zhang. 2012. *NDNLP: A Link Protocol for NDN.* Technical Report. NDN-0006, Revision 1. https://named-data.net/publications/techreports/trlinkprotocol/

[42] Chrome DevTools team. 2023. *Puppeteer*. Retrieved June 12, 2023 from https://pptr.dev/

[43] Rama Krishna Thelagathoti, Spyridon Mastorakis, Anant Shah, Harkeerat Bedi, and Susmit Shannigrahi. 2020. Named Data Networking for Content Delivery Network Workflows. https://doi.org/10.1109/CLOUDNET51028.2020.9335806

[44] Michael Tüxen, Fulvio Risso, Jasper Bongertz, Gerald Combs, Guy Harris, Eelco Chaudron, and Michael Richardson. 2023. *PCAP Next Generation (pcapng) Capture File Format*. Internet-Draft draft-ietf-opsawg-pcapng-01. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-ietf-opsawg-pcapng/01/ Work in Progress.

[45] Yuanhao Wu, Faruk Volkan Mutlu, Yuezhou Liu, Edmund Yeh, Ran Liu, Catalin Iordache, Justas Balcas, Harvey Newman, Raimondas Sirvinskas, Michael Lo, Sichen Song, Jason Cong, Lixia Zhang, Sankalpa Timilsina, Susmit Shannigrahi, Chengyu Fan, Davide Pesavento, Junxiao Shi, and Lotfi Benmohamed. 2022. N-DISE: NDN-Based Data Distribution for Large-Scale Data-Intensive Science. In *Proceedings of the 9th ACM Conference on Information-Centric Networking (ICN '22)*. Association for Computing Machinery, 103–113. https://doi.org/10.1145/3517212.3558087

[46] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, KC Claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and Beichuan Zhang. 2014. Named Data Networking. *ACM SIGCOMM Computer Communication Review* 44, 3 (2014), 66–73.

[47] Minsheng Zhang, Vince Lehman, and Lan Wang. 2017. Scalable name-based data synchronization for named data networking. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*. 1–9. https://doi.org/10.1109/INFOCOM.2017.8057193

[48] Zhiyi Zhang, Edward Lu, Yanbiao Li, Lixia Zhang, Tianyuan Yu, Davide Pesavento, Junxiao Shi, and Lotfi Benmohamed. 2018. NDNoT: A Framework for Named Data Network of Things. In *Proceedings of the 5th ACM Conference on Information-Centric Networking (ICN '18)*. Association for Computing Machinery, 200–201. https://doi.org/10.1145/3267955.3269019

Appendix A DATASET AND TOOLS

This work is part of a continuous data collection effort. The latest version of these tools and the continuously growing set of traffic traces is available at: **https://www.tntech-ngin.net/datasets**. All artifacts produced in this work are open-source and publicly available. See the table below for details.

Table 3: Links to the full dataset of traces and the tools we used at the time of writing this paper.

URL	Description
https://github.com/tntech-ngin/ndn-traffic-traces	Full set of anonymized and compressed packet traces
https://github.com/usnistgov/ndntdump	General-purpose NDN traffic dumper
https://github.com/tntech-ngin/ndn-traffic-capture-scripts	Scripts to automatically capture traffic with *ndntdump*
https://github.com/tntech-ngin/ndn-traffic-plotting-scripts	Various plotting tools for NDN packet traces
https://github.com/tntech-ngin/ndn-traffic-replay-ndnSIM	Sample application to replay packet traces in ndnSIM